1976-06

A study of pressure-volume rates and plenum membrane additions to the captured air bubble surface effect ship XR-3 digital computer loads and motion program

Boggio, John Martin
Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/17964

Downloaded from NPS Archive: Calhoun
A STUDY OF PRESSURE-VOLUME RATES AND PLENUM MEMBRANE ADDITIONS TO THE CAPTURED AIR BUBBLE SURFACE EFFECT SHIP XR-3 DIGITAL COMPUTER LOADS AND MOTION PROGRAM

John Martin Boggio
THESIS

A STUDY OF PRESSURE-VOLUME RATES AND PLENUM MEMBRANE ADDITIONS TO THE CAPTURED AIR BUBBLE SURFACE EFFECT SHIP XR-3 DIGITAL COMPUTER LOADS AND MOTION PROGRAM

by

John Martin Boggio

June 1976

Thesis Advisor: A. Gerba, Jr.

Approved for public release; distribution unlimited.
A study was conducted of modeling changes to the XR-3 Captured Air Bubble Surface Effect Ship digital computer Loads and Motion program. These changes included the addition of pressure rate and volume rate to the existing 6 degrees of freedom equations. Additional equations were developed to simulate the application of a nonpermeable membrane to the plenum of the XR-3 test craft. The objectives of this study were to determine whether the
addition of pressure rate and volume rate equations would improve computer execution time and to test some simplified models of the plenum membrane. Computer timing improvements were demonstrated and membrane modeling results are presented.
A Study of Pressure-Volume Rates and Plenum Membrane Additions to the Captured Air Bubble Surface Effect Ship XR-3 Digital Computer Loads and Motion Program

by

John Martin Boggio
Lieutenant, United States Navy
B.S. E.E., Purdue University, 1967

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1976
ABSTRACT

A study was conducted of modeling changes to the XR-3 Captured Air Bubble Surface Effect Ship digital computer Loads and Motion program. These changes included the addition of pressure rate and volume rate to the existing 6 degrees of freedom equations. Additional equations were developed to simulate the application of a nonpermeable membrane to the plenum of the XR-3 test craft. The objectives of this study were to determine whether the addition of pressure rate and volume rate equations would improve computer execution time and to test some simplified models of the plenum membrane. Computer timing improvements were demonstrated and membrane modeling results are presented.
TABLE OF CONTENTS

I. INTRODUCTION .................................................. 10
   A. BACKGROUND ............................................. 10
   B. OBJECTIVES ............................................ 11

II. PROGRAMMING CHANGES .......................................... 12
    A. RATE EQUATIONS FOR PRESSURE AND VOLUME .......... 12
       1. Introduction ........................................ 12
       2. SUBROUTINES RHS and INTGRL ...................... 12
       3. Formulation of Volume Rate and Pressure Rate .. 13
    B. ELIMINATION OF SUBROUTINE DMINV ..................... 16
    C. EXPANSION OF WAVES COMPONENTS ..................... 18
    D. MISCELLANEOUS ....................................... 18

III. TEST PROCEDURES ............................................... 20
    A. COMPUTER SYSTEM ..................................... 20
    B. TEST CONDITIONS .................................... 21

IV. MEMBRANE ADDITION ........................................... 25
    A. INTRODUCTION ......................................... 25
    B. MATHEMATICAL MODELING ............................... 25
       1. Model One ......................................... 25
       2. Model Two ......................................... 26
       3. Model Three ....................................... 26
       4. Model Four ........................................ 28
       5. Model Five ....................................... 31

V. DISCUSSION OF RESULTS ........................................ 32
    A. PRESSURE AND VOLUME RATE CHANGES ................. 32
<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I. TEST PERTURBATIONS</td>
<td>22</td>
</tr>
<tr>
<td>II. STEADY STATE CONDITIONS</td>
<td>24</td>
</tr>
<tr>
<td>III. TIMING QUALITY FACTORS 20 KNOTS</td>
<td>33</td>
</tr>
<tr>
<td>IV. TIMING QUALITY FACTORS 30 KNOTS</td>
<td>33</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1. Computational Flow Diagram Model I --------------- 38
2. Computational Flow Diagram Model II --------------- 39
3. Computational Flow Diagram Model III --------------- 40
4. Computational Flow Diagram Model IV --------------- 41
5. Computational Flow Diagram Model V --------------- 42
6. Center of Gravity Accelerations with Membrane Sea State One --------------------------------- 43
7. Center of Gravity Acceleration without Membrane Sea State One ----------------------------- 44
8. Center of Gravity Acceleration with Membrane Sea State Two ----------------------------- 45
9. Center of Gravity Acceleration without Membrane Sea State Two ----------------------------- 46
ACKNOWLEDGEMENT

The author wishes to express his appreciation to Associate Professor Alex Gerba, of the U. S. Naval Postgraduate School, for the motivation, guidance, encouragement and above all, the patience which he provided during the course of this work.

The author also wishes to express his appreciation to his wife for her encouragement and support plus the many hours of typing which transformed some indecipherable symbols into a legible manuscript.
I. INTRODUCTION

A. BACKGROUND

The Captured Air Bubble Surface Effect Ship (CAB SES) offers a dynamic new approach to improve surface ship performance. As with any new approach, extensive testing at the design stage of development is required. The U.S. Navy has been conducting sea trials on two 100 ton Captured Air Bubble (CAB) test craft, the 100-A and the 100-B Ref. 1. In addition to scale model testing, another approach to design evaluation involves the use of high speed digital computers to simulate the dynamic performance of the craft.

A digital computer Loads and Motion (L&M) program for the CAB SES was developed under government contract by the Oceanics Corp. Ref. 2. The L&M program installed at the Naval Postgraduate School, W. R. Church, computer facility on an IBM 360/67 computer was for a 100 ton displacement test craft, the 100-B. Leo and Boncal Ref. 3 modified this computer program to represent a smaller 3 ton craft, the XR-3. The XR-3 is an operational test craft currently maintained and operated by the Naval Postgraduate School Ref. 4.

Various modifications and changes to Leo and Boncal's basic work have been implemented. Finley, Forbes, and Menzel, in Refs. 5-7, provides program changes to obtain better representation of the Fan maps, bow and stern seal dynamics and pitch and roll damping.
One adverse characteristic of both the 100-B computer program and the XR-3 computer program was the extensive use of the digital computer time when operating in simulated sea states. Since the 100-B and the XR-3 programs are similar, the computer time analysis provided by Mitchell \( \text{Ref. 8}\) for the 100-B also applies to the XR-3 program.

Another aid to design is the testing done in towing tanks in which scale models are used. Towing tests were conducted at NSRDC using a rubber membrane installed on the forward portion of a CAB SES model in order to better scale the pressure-volume relationship \( \text{Ref. 9}\). The addition of this membrane improved the pitch and heave characteristics of the scale model in sea state operation.

B. OBJECTIVES

It is the objective of this work to examine the L&M digital computer program and make any changes that would improve the program with particular attention to changes that would reduce the amount of time required to execute the program.

A second objective is to model the plenum membrane, add the required program changes and study the effects of the membrane on simulated craft performance.
II. PROGRAMMING CHANGES

A. RATE EQUATIONS FOR PRESSURE AND VOLUME

1. Introduction

An examination of the dynamics of the XR-3 L&M program indicate that the key parameters of this type of ship design are those which describe the dynamic behavior of the bubble of the air in the plenum. This pressure of air provides approximately 75% of the lift force when the craft is "on the bubble." Small changes in bubble volume and pressure therefore cause significant changes in total ship response. These variables were examined in an effort to decrease the digital computer simulation execution time.

2. SUBROUTINES RHS and INTGRL

The force and moment equations for the craft are the six degrees of freedom equations plus four auxiliary equations contained in SUBROUTINE RHS [Ref. 2].

The integration of these equations is performed in SUBROUTINE INTGRL. INTGRL uses the Runge-Kutta-Merson numerical integration technique with an automatic variable step size. If the error tolerances specified are exceeded the step size is reduced and the step repeated until all error tolerances are met or the step size is reduced to \(1 \times 10^{-6}\) sec., at which point the program stops. Thus the integration values are forced to converge to their correct values. All ten equations are integrated serially at the
same time step. When all integration error tolerances are met INTGRL returns the integrated values and increases the simulation time one time-step increment. Since pressure and volume are key variables, integrating their rates should cause the system of equations to converge more rapidly to their true values and avoid any problems associated with the feedback loops in the pressure-volume calculation \cite{Ref. 10}.

3. **Formulation of Volume Rate and Pressure Rate**

The bubble pressure equation is for an adiabatic process \cite{Ref. 2}.

\begin{equation}
  P_b = P_a \left( \frac{\rho_b}{\rho_a} \right)^\gamma
\end{equation}

where the subscripts \(a\) and \(b\) represent atmosphere and plenum values respectively.

\( P = \) pressure absolute

\( \rho = \) density \( \frac{M}{V} \)

\( \gamma = \) adiabatic exponent for air

Substituting (2) into (1)

\begin{equation}
  P_b = P_a \left( \frac{\frac{M}{V \rho_a}}{\rho_a} \right)^\gamma
\end{equation}

Differentiating (3) with respect to time where \( P_a \) and \( \rho_a \) are assumed to be constant yields

\begin{equation}
  \dot{P}_b = \frac{P_a}{\rho_a} \gamma \left( \frac{\frac{M}{V \rho_a}}{\rho_a} \right)^{\gamma-1} \left( \frac{\dot{V}_M - \dot{M}_V}{V^2} \right)
\end{equation}
Factoring $\frac{M}{V}$, from the last term of (4)

$$\dot{P}_b = \gamma P_a \left( \frac{M}{\rho_a V} \right)^{\gamma-1} \left( \frac{M}{\rho_a V} \right) \left( \frac{\dot{M}}{M} - \frac{\dot{V}}{V} \right)$$

(5)

Noting \(\left( \frac{M}{\rho_a V} \right)^{\gamma-1} \left( \frac{M}{\rho_a V} \right) = \left( \frac{M}{\rho_a V} \right)^\gamma\)

and

$$P_a \left( \frac{M}{\rho_a V} \right)^\gamma = P_b$$

$$\dot{P}_b = \gamma P_b \left( \frac{\dot{M}}{M} - \frac{\dot{V}}{V} \right)$$

(6)

The volume equation for the bubble is computed in RHS in four parts and summed together

$$V = V_{nom} - D - WA + WT$$

(7)

The four terms that determine volume are

(1) $V_{nom}$ = the empty plenum volume
(2) $D$ = draft term--change in volume due to draft
(3) $WA$ = Wave term--change in volume due to the presence of waves
(4) $WT$ = WATSLP term-- correction term for the added volume wedge which is a function of speed \(\dot{V}\) \(\boxRef{6}\).

Differentiating equation (4)

$$\dot{V} = -\dot{D} - \dot{WA}$$

(8)
where

\[ \dot{\text{WA}} = \text{rate of change of volume due to waves} \]
\[ D = \text{rate of change of volume due to displacement} \]
\[ \dot{\text{WT}} = 0.0 \]
\[ \text{Vnom} = 0.0 \]

\( \dot{\text{WA}} \) is computed in SUBROUTINE WAVES by taking the difference in change of volume due to waves at successive time steps and dividing by the time step. Since \( \dot{\text{WATSLP}} \) is a function of speed, the differentiated value of \( \dot{\text{WATSLP}} \), \( \dot{\text{WT}} \), is a function of acceleration. Forward accelerations are normally small; therefore, the differentiated value of \( \dot{\text{WATSLP}} \) was approximated to be zero. Equations (4) and (5) were programmed as follows:

\[
\text{VALUE (11)} = -(((\text{XL} \times \text{WIDTH}) - \text{ABW}) \times 0.5 \times \text{W}) - \text{DVWDOT}
\]
\[
\text{VALUE (12)} = \text{GAM} \times \text{VAL(13)} \times (\text{VALUE(10)} / \text{VAL(11)} - \text{VALUE(11)} / \text{VAL(12)})
\]

where

\( \text{DVWDOT} \) = rate of change of volume due to waves
\( \text{VALUE(10)} \) = Bubble mass flow rate
\( \text{VALUE(11)} \) = Volume rate
\( \text{VALUE(12)} \) = Pressure rate
\( \text{VAL(11)} \) = Bubble Mass
\( \text{VAL(12)} \) = Bubble Volume
\( \text{VAL(13)} \) = Bubble Pressure
\( \text{GAM} \) = Adiabatic Exponent for air
B. ELIMINATION OF SUBROUTINE DMINV

In an effort to improve program timing, the suggestions given in Ref. 7 were undertaken. One recommendation suggested the removal of SUBROUTINE DMINV in the 100B L&M program by making changes to SUBROUTINES INCON and RHS. An examination of the L&M program indicated that these changes also could be applied to the XR-3 program. Consequently the following was deleted:

```
212 DO 211 I = 1,6
     DO 211 N = 1,6
211 A(I,N) = 0.0
     DO 213 N = 1,3
213 A(N,N) = AM
     A(4,4) = AIXX
     A(5,5) = AIYY
     A(6,6) = AIZZ
     A(4,6) = -AIXZ
     A(6,4) = -AIXZ
     AIMAX = AMAXI(AM, AIXX, AIYY, AIZZ, ABS(AIXZ))
     DO 214 I = 1,6
     DO 214 J = 1,6
214 A(I,J) = A(I,J)/AIMAX
     CALL DMINV(A,G,D,)
     DO 215 I = 1,6
215 A(I,J) = A(I,J)/AIMAX
     IF (D.NE.0.0.) GO TO 10
     WRITE (6,216)
     STOP
```
In place of the above the following was added:

215  AMASSI = 1.0/AM

\[ D = 1.0 / (AIXX*AIYZ - AIXZ*AIYZ) \]

\[ DIXX = AIXX*D \]
\[ DIXZ = AIYZ*D \]
\[ DIYZ = AIYZ*D \]
\[ AIYY = 1.0/AIYY \]

GO TO 10

Linkage between INCON and RHS was provided by the following

COMMON/ATRIX/AMASSI, AIYY, DIXX, DIXZ, DIYZ

In SUBROUTINE RHS the six element matrix GF(J) was deleted and the following identifiers were substituted for the summation of forces: SUMX, SUMY, SUMZ, SUMK, SUMM, SUMN. In addition, the following deletion was made.

\[ \text{DO } 1 \ I = 1,6 \]
\[ \text{VALUE}(I) = 0.0 \]

\[ \text{DO } 1 \ J = 1,6 \]
\[ \text{VALUE} (I) = \text{VALUE} (I) + A(I,J)*GF(J) \]

1 CONTINUE

Substituted for the above DO LOOPS was the following

\[ \text{VALUE} (1) = \text{SUMX*AMASSI} \]
\[ \text{VALUE} (2) = \text{SUMY*AMASSI} - R*U \]
\[ \text{VALUE} (3) = \text{SUMZ*AMASSI} \]
\[ \text{VALUE} (4) = \text{SUMK*DIYZ} + \text{SUMN*DIXZ} \]
VALUE (5) = SUMM*AIYYI
VALUE (6) = SUMN*DIXX + SUMK*DIXZ

These changes eliminated SUBROUTINE DMINV and several DO LOOPS.

C. EXPANSION OF WAVE COMPONENTS

Since the program was being tested with sea state, the method used for the wave generation was studied. SUBROUTINE INCON provides a means of introducing sea-state by individual wave components and amplitudes. In addition, the subroutine will accept an average height and lowest and highest wave frequency and/or wave period and generate up to 10 wave components.

These wave components represent a sampling of the spectrual energy density of the given sea state condition. Reference 11 suggests for irregular sea, a minimum of 15 - 20 components are required to simulate irregular sea conditions. The program dimension statements were accordingly increased from 10-20 to determine if any significant changes could be observed by increasing the number of wave components.

D. MISCELLANEOUS

Each subroutine was examined with regard to efficiency of coding and changes were implemented such as multiplication in lieu of the use of the exponential. For example, IBM 360/67 fixed point exponentiation requires that the natural logarithm be generated and then a logarithm be computed, as compared
to straightforward multiplication. When such exponentiation is nested in DO LOOPS, a significant time savings can be realized by the use of multiplication.
III. TEST PRODECURES

A. COMPUTER SYSTEM

The digital computer used throughout this study is an IBM 360/67 model, VERSION I, located at the NPS W. R. Church Computer Facility. The system's hardware configuration provides 2 M bytes of core. This main core is composed of IBM core storage devices (IBM 2365 Mod. 12) and compatible but slightly different Lockheed core storage devices (MM 365). The Lockheed core devices are about 18% slower in execution time than IBM core devices. Since the system is time-shared and the core is contiguous, there is no practical way to determine in which portion of the main core the program is residing. Therefore, exact timing of a program is not easily achieved. In this report all timing values must be interpreted with this anomaly in mind. Among the options available under this operating system are the FORTRAM G and FORTRAN H compilers [Ref. 12]. Mitchell reported substantial time savings using the H compiler [Ref. 8]. Since one of the main objectives was to reduce execution time, computer model testing was done using both compiler options. Runs made under different compilers will be noted.

A word of caution on the use of the H compiler. The H compiler produces its timing improvements and core reduction by coding optimization. This means the compiler re-arranges
the Fortran code. This re-arrangement may move computations outside of DO LOOPS for example. Applicable portions of Ref. 13 should be read and understood before using this feature.

B. TEST CONDITIONS

The L&M program for the XR-3 used in this report is the one given in Ref. 7. This program was then modified to incorporate the changes previously described. The basic program [Ref. 7] computes pressure using eq. (1) and volume using eq. (7) and is referred to as PROGRAM ONE. PROGRAM ONE was used as reference for comparative purposes. The addition of the pressure rate and volume rate equations (6) and (8) and the elimination of SUBROUTINE DMINV constituted major modifications to the basic program. This version of the Loads and Motion program was called PROGRAM TWO. PROGRAM THREE was PROGRAM TWO compiled and executed with the FORTRAN H compiler.

Five test conditions for a given speed were established to provide a full range of disturbances for testing the L&M programs. These test conditions are tabulated in Table I.

In Condition One small disturbances were introduced by changing the initial conditions of draft (DS) and pitch (θ). These changes consisted of decreasing the draft and pitch from previously computed steady state values by the magnitude shown in Table I. In addition, the rudder was held at zero and the sea was calm. In Condition Two, the perturbations were the same as Condition One except that now the rudder
### TABLE I

**TEST PERTURBATIONS**

<table>
<thead>
<tr>
<th>Condition</th>
<th>ΔDS</th>
<th>Δθ</th>
<th>RUDDER</th>
<th>SEA STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>0.1 inch</td>
<td>0.03</td>
<td>0.0</td>
<td>Calm water</td>
</tr>
<tr>
<td>Two</td>
<td>0.1</td>
<td>0.03</td>
<td>20°</td>
<td>Calm water</td>
</tr>
<tr>
<td>Three</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>Regular One</td>
</tr>
<tr>
<td>Four</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>Regular Two</td>
</tr>
<tr>
<td>Five</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>Irregular One</td>
</tr>
</tbody>
</table>

was displaced to the right 20 degrees. The rudder was offset at the start of the simulation and held through the run as this value. In Condition Three, Four and Five the only disturbances used were the sea state conditions.

Sea state is generated in SUBROUTINE WAVES. SUBROUTINE INC0N generates wave components and amplitudes based on several different type input parameters. See users manual Ref. 7. Regular sea one and two are simple sinusoidal waves. Irregular sea state conditions are simulated by the addition of several regular sinusoidal wave components. Sea state used throughout this study is true sea state and is not scaled in any manner.

The L&M program provides two options for propulsion, either constant speed or constant thrust. The constant...
thrust option was selected because this is the normal operating mode for the XR-3 test craft.

Integrator tolerance levels were maintained at the values given in Ref. 7. However two new error tolerances had to be determined for the integration of the pressure and volume rates. These new levels were determined by first selecting a very tight value .000001 which was increased in increments until a change in output could be noticed. The value was then decreased by a factor of 10. Pressure and volume tolerance levels were then set at this value which was found to be .0001.

All other variables such as draft and pressure were initialized from the steady state undisturbed conditions for that specific starting speed. The steady state conditions were evaluated by first using the constant speed propulsion option and the initial values in Ref. 7. The program was then run until the key input variables - pitch, draft, thrust and pressure - reached steady state. These values were then read in and propulsion was switched to constant thrust. The program was run again to insure that steady state conditions existed. These steady state values are shown in Table II.
<table>
<thead>
<tr>
<th>Speed (Knots)</th>
<th>10</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure</td>
<td>23.93</td>
<td>24.84</td>
<td>24.84</td>
</tr>
<tr>
<td>Draft</td>
<td>8.17</td>
<td>6.12</td>
<td>5.34</td>
</tr>
<tr>
<td>Thrust (each engine)</td>
<td>200.31</td>
<td>218.17</td>
<td>287.22</td>
</tr>
<tr>
<td>Pitch</td>
<td>1.62</td>
<td>.48</td>
<td>.26</td>
</tr>
</tbody>
</table>

**Pressure in psfg**  
**Draft in inches**  
**Thrust in ft. lb.**  
**Pitch in degrees**
IV. MEMBRANE ADDITION

A. INTRODUCTION

During towing tank tests conducted at NSRDC, a rubber membrane was installed on the forward bow of the XR-1 scale model [Ref. 9]. The installation of this membrane improved the pitch and heave characteristics of the scale model during simulated sea conditions. Reference 9 provided the following empirical expression which relates the Pressure and Volume of the membrane.

\[ V = KP_b \] (9)

K was experimentally determined and has the value .317 when scaled to the XR-3 test craft dimensions. (See Appendix A).

B. MATHEMATICAL MODELING

The approach used to include the effects of this membrane was to develop mathematical expressions which could be added to the already existing pressure, volume and air mass equations in the L&M program. Five mathematical expressions were developed, tested and referred to as Models One, Two, Three, Four and Five. They are explained below and the results discussed in Section V.

1. Model One

Model One is based on the premise that by correcting the plenum volume with the membrane volume, the equations
in SUBROUTINE RHS would show the effect of the membrane. Model One added the following equation to PROGRAM ONE;

\[ V_m = K P_m \] (10)

Then using (10) to correct the volume of the plenum and assuming \( P_m = P_b \).

\[ V = V_b + V_m \] (11)

where \( V_b \) is given by eq. (7).

2. Model Two

Model Two is an extension of Model One. Based on the premise that by correcting the rate of change of plenum volume with the rate of change of membrane volume, the rate equations given in SUBROUTINE RHS and subsequent integration of these terms would show the effect of the membrane. Model Two added the following rate equation to PROGRAM TWO.

\[ \dot{V}_m = K \dot{P}_m \] (12)

Then using (12) to correct the rate of change of plenum volume and assuming

\[ \dot{P}_m = \dot{P}_b \]

\[ \dot{V} = \dot{V}_b + \dot{V}_m \]

where \( \dot{V}_b \) is given by eq. (8).

3. Model Three

Model Three was based on the premise that the volume rate eq. (8) and the pressure rate eq. (6) as computed without the membrane could be corrected for the membrane by adjusting these terms by appropriate factors within PROGRAM TWO.
Correction terms were sought that would be a function of the membrane. A logical candidate for this correction factor for the pressure was the ratio of rates of change of the membrane volume to plenum volume multiplied by plenum pressure rate.

\[ PCT = \frac{\dot{V}_m}{V_b} \frac{\dot{P}_b}{\dot{P}_b} \] (13)

where PCT is the correction factor to be added to the plenum pressure rate term \( \dot{P}_b \) before integration. It was assumed that the pressure and pressure rate of the plenum and membrane were equal.

\[ P_m = P_b \] (14)
\[ \dot{P}_m = \dot{P}_b \] (15)

It was also assumed that the membrane volume rate and pressure rate were related by

\[ \dot{V}_m = K \dot{P}_m \] (16)

Substituting (16) into (13) and using (15)

\[ PCT = K \frac{\dot{P}_b^2}{V_b} \] (17)

The volume rate correction term was assumed to be the ratio of the rate of change of membrane pressure to the rate of change of plenum pressure times the rate of change of membrane volume.
\[ VCT = \frac{\dot{P}_m}{\dot{P}_b} \dot{V}_m \]  

(18)

where \( VCT \) is the correction factor to be added to the plenum volume rate term \( \dot{V}_b \) before integration.

Since \( \dot{P}_m = \dot{P}_b \) and \( \dot{V}_m = K \dot{V}_b \)

Then from (18)

\[ VCT = K \dot{P}_b \]  

(19)

Then using equations (17) and (19) the rate equations are

\[ \dot{V} = \dot{V}_b + VCT \]  

(20)

\[ \dot{P} = \dot{P}_b + PCT \]  

(21)

where \( \dot{V}_b \) is given by eq. (8) and \( \dot{P}_b \) by eq. (6).

Consider the following. Since \( K \) is a function of the elasticity of the membrane, if \( K = 0.0 \). Equations (20) and (21) reduce to the no membrane case.

If the membrane expands at the same rate that the volume of the plenum contracts the system volume rates is zero and the system pressure rate is also zero.

4. Model Four

Again using PROGRAM TWO, Model Four was based on the premise that the mass of the membrane air and mass flow rate of membrane air would be important factors in any correction term to be applied to plenum volume rate and pressure rate.
Again it was assumed that the membrane effect could be described as an adiabatic reversible process. Then using Equation (6) applied to the membrane,

\[ P_m = P_m \gamma \left( \frac{\dot{M}_m}{M_m} - \frac{\dot{V}_m}{V_m} \right) \]  

(22)

where \( m \) denotes membrane

\[ \dot{M}_m = \text{Mass flow rate of the membrane air} \]
\[ M_m = \text{Mass of the membrane air} \]

Again it was assumed that the pressure and pressure rate of the membrane and plenum were equal.

\[ P_m = P_b \]  

(23)
\[ \dot{P}_m = \dot{P}_b \]  

(24)

It was also assumed that the membrane volume rate and pressure rate were related by

\[ \dot{V}_m = K \dot{P}_m \]  

(25)

Equation (25) was obtained by differentiating

\[ V_m = K P_m \]  

(26)

Substituting (25) and (26) into (22) yields

\[ \frac{M_m}{M_m} = \left( \frac{1+\gamma}{\gamma} \right) \frac{\dot{P}_m}{P_m} \]  

(27)
One term is not available, \( M_m \). Two possibilities were considered to determine this value. The first possibility would be to integrate \( M_m \) and use the result of the integration from the preceding time step. The other possibility would be to hold temperatures constant in the membrane, and use the ideal gas law to determine \( M_m \) for varying pressure and volume. This last condition would be in error; however, as a first approximation, it would produce acceptable results.

Because of the past history of difficulty with integrator 10, plenum air mass rate, the second possibility was chosen. Evaluation of the terms in the ideal gas law for a temperature of 68° degrees excluding mass, pressure and volume produced the constant.

\[ 4.9 \times 10^{-7} \text{ slugs/psf} - \text{ft}^3 \]

therefore;

\[ M_m = (4.9 \times 10^{-7})(P_m)(V_m) \quad (28) \]

Equations (22), (25) and (27) describe the membrane. These equations are used to correct the mass rate and volume rate of the plenum without the membrane.

\[ \dot{M}_s = \dot{M} + \dot{M}_m \quad (29) \]

\[ \dot{V}_s = \dot{V} + \dot{V}_m \quad (30) \]

where

\( s \) = denotes the system of plenum plus membrane

\( \dot{M} \) = mass flow rate of plenum

\( \dot{V} \) = volume rate of change of plenum
Using (29) and (30) in equation (6) the corrected plenum pressure rate is

\[ \dot{P}_s = P_b \gamma \left( \frac{\dot{M}_s}{M} - \frac{\dot{V}_s}{V} \right) \]

where \( P_b, M \) and \( V \) are values obtained from the previous time-step calculation.

5. **Model Five**

Model Five is the application of equation (10) to PROGRAM TWO. It is based on the premise that the volume of the membrane could be introduced into the plenum pressure rate equation, and represent the membrane effect.

\[ V_m = K P_m \]  \hspace{1cm} (31)

\[ V = V_b + V_m \]  \hspace{1cm} (32)

Equation (31) and (32) are computed as in Model One. Then equation (32) is used in the pressure rate equation.

\[ \dot{P} = P_b \gamma \left( \frac{\dot{M}}{M} - \frac{\dot{V}}{V} \right) \]  \hspace{1cm} (33)
V. DISCUSSION OF RESULTS

A. PRESSURE RATE AND VOLUME RATE CHANGE

The addition of the pressure rate and volume rate equations was implemented without any major difficulty. In order to provide a basis of comparison for computer time analysis a quality factor "Q" was established. "Q" is defined to be the ratio of CPU execution time to problem time. For example, if the problem specified 1 minute of simulation and the CPU execution time (GO STEP in IBM COMPUTER SYSTEMS) was 5 minutes Q = 5.0. PROGRAMS ONE, TWO and THREE were tested with identical input parameters. The results are summarized in Table III and Table IV.

Runs were compared using PROGRAM ONE as a reference. No differences in computed output were noted for Condition One thru Four and only small changes in computed values noted with Condition Five. The integrator error tolerance levels were maintained constant for all runs.

As can be seen from the tabulated values, the addition of pressure rate and volume rate equations reduces the CPU execution time considerably for small disturbances. However, as the sea state is increased the CPU time increases. Apparently other factors in the program begin to dominate the total execution time.
### TABLE III

**TIMING QUALITY FACTORS**

**20 Knots**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Program One</th>
<th>Program Two</th>
<th>Program Three</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition One</td>
<td>11.25</td>
<td>4.8</td>
<td>4.50</td>
</tr>
<tr>
<td>Condition Two</td>
<td>13.85</td>
<td>6.0</td>
<td>5.50</td>
</tr>
<tr>
<td>Condition Three</td>
<td>23.0</td>
<td>20.5</td>
<td>19.2</td>
</tr>
<tr>
<td>Condition Four</td>
<td>42.8</td>
<td>28.01</td>
<td>22.0</td>
</tr>
<tr>
<td>Condition Five</td>
<td>71.0</td>
<td>58.1</td>
<td>54.0</td>
</tr>
</tbody>
</table>

### TABLE IV

**TIMING QUALITY FACTORS**

**30 Knots**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Program One</th>
<th>Program Two</th>
<th>Program Three</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition One</td>
<td>10.8</td>
<td>7.5</td>
<td>4.5</td>
</tr>
<tr>
<td>Condition Two</td>
<td>18.0</td>
<td>14.25</td>
<td>8.7</td>
</tr>
<tr>
<td>Condition Three</td>
<td>24.0</td>
<td>22.0</td>
<td>21.0</td>
</tr>
<tr>
<td>Condition Four</td>
<td>45.0</td>
<td>30.0</td>
<td>28.0</td>
</tr>
<tr>
<td>Condition Five</td>
<td>75.0</td>
<td>59.2</td>
<td>53.0</td>
</tr>
</tbody>
</table>
This supports the results of Reference (?) that for sea conditions the bulk of computation time is apparently in the SUBROUTINE WAVES.

It was observed that at irregular sea state two and regular sea state three water came in contact with the top of the plenum. The L&M program does not include the effect of this phenomenon in the program simulation; therefore, testing was not conducted at higher sea state conditions. The fact that water hits the top of the plenum is a reasonable condition under certain circumstances. For example, if the pitch angle is 3 degrees up the stern sinks over 6 inches. If the draft of the craft at that time is 11 inches and if the preceding wave passing the stern station is 5 inches, water should hit the top of the plenum at the stern station. The effect of this condition on total craft behavior in terms of forces and moments is not known. The effect of water contact would depend upon such factors as the length of time the wave is in contact with the top of the plenum and the area of the plenum in contact with the wave.

B. INCREASING WAVE COMPONENTS

Tests were made for irregular seas one and two using 8, 10, 15, and 20 wave components. No apparent change in output could be determined. The time at which changes in output occurred changed, but no significant reduction in output values were noted; that is the time when wave components were in-phase or out-of-phase was different, but the craft response
was the same. It was felt that if more components were used, the energy (in this case, the magnitude and frequency of the disturbing force) would be distributed over many components and that the irregular combination of these components would spread the energy of the disturbance over a larger time base. This spreading of the energy would then reduce the magnitude of the disturbance and that in turn would provide less rapid rates of change. With less rapid changes in the disturbances, the integration would take less time. The over-all time increase due to more wave components would be offset by the reduction in integration time. This is not the case; using more wave components increased CPU time. Whether 10 or less wave components are sufficient to accurately represent irregular seas was not tested.

C. MEMBRANE MODELS

A large amount of time was expended in attempting to provide adequate modeling of the plenum membrane. Model One using \[ V_m = K P_b \] in PROGRAM ONE did not work at all. The program would begin to execute and then stop when the integration step size was less than \[ 1 \times 10^{-6} \].

As an aid to find the reason for this failure a computational flow diagram was drawn (See Figure 1). The addition of the correction term \( V_m \) creates an algebraic inner loop as shown by the dashed line in Fig. 1. Using a small perturbation of \( D \) in calm water and using the print switches in RHS as well as using the DEBUG option available with the G compiler, it
was observed that with Model One the values of pressure were divergent. It is not known why pressure values diverge; however, it was noted that at certain times the values of pressure and volume were increasing or decreasing together, that is when pressure was increasing while volume was increasing.

The computational flow diagram for Model Two (Figure 2) shows the inner connections of the pressure, volume and mass variables. Model Two exhibited the same symptoms and the same results as Model One, except in some cases pressure rate and volume rate rose or fell together at the same time step.

The failures of Model One and Two led to Model Three. It was felt that by correcting both pressure rate and volume rate the divergence exhibited in Models One and Two could be corrected. Model Three also failed; however, it provided the motivation to consider mass rate into the membrane. The computations performed in Model Three is shown in Figure 3.

Model Four was marginally successful. The program executes and slightly reduces the center of gravity acceleration. The reduction of the center of gravity acceleration was chosen as the criterion of whether the model was representing the membrane. This model was tested in sea state one and sea state two and provides slight reduction in the maximum values in the center of gravity acceleration. It did not reduce all of the center of gravity values. The computational flow diagram
of Model Four is given in Figure 4 and shows no algebraic loops in the RHS computations of pressure and volume rates.

Model Five was the most successful model. The computational flow is shown in Figure 5. Model Five reduced the center of gravity acceleration and for small disturbances, the pitch angle and pitch rate. For sea state one a significant reduction was observed. (See Figures 6 and 7). However, for sea state two the center of gravity accelerations were greater for the membrane when compared to the no membrane case. (See Figures 8 and 9). No reason for this could be found. The only differences in these two runs was the sea state.

One possible reason for the sea state two results, though not yet proven, could be a resonance effect at the sea state encounter frequency. Concurrent work in this area [Ref. 14] investigates the conditions for the existence of any resonance phenomena. Preliminary investigations indicate that at a wave encounter frequency of approximately 4 rad / second resonance in pitch angle occurs. The encounter frequency for regular sea state two is 5 rad / second. Whether resonance phenomena could cause this effect, however, was not determined.
(All values shown for the $K^{th}$ iteration in RHS)

Figure 1. Computational Flow Diagram Model I
Figure 2. Computational Flow Diagram Model II
Figure 3. Computational Flow Diagram Model III
\[
\begin{align*}
\dot{P}_m^k & \rightarrow K \rightarrow v_m^k \rightarrow \dot{P}_m^k \rightarrow \text{CCF} \\
\dot{P}_m^k & \rightarrow \frac{1}{\gamma} \rightarrow \frac{\gamma+1}{\gamma} \rightarrow \sum M_m^k \rightarrow \sum M_s^k \rightarrow \dot{M} \\
\sum & \rightarrow \frac{M_s^k}{M_s} \rightarrow \frac{M_s^k}{M_s} \rightarrow \dot{P}_s \rightarrow \int \rightarrow P^{k+1} \\
\dot{P}_b^k & \rightarrow K \rightarrow v_m^k \rightarrow \sum v_s^k \rightarrow v^k \rightarrow v^k
\end{align*}
\]

CCF = \(4.9 \times 10^{-7}\)

Figure 4. Computational Flow Diagram Model IV
Figure 5. Computational Flow Diagram Model V
FIGURE 6

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=5.00E-02 UNITS INCH.

PROGRAM 2 MODEL 5 20KTS REG. SEA 1
PLOT IS C.G. ACCELERATION VERSUS TIME
FIGURE 7

X-SCALE = 5.00E+00 UNITS INCH,
Y-SCALE = 5.00E-02 UNITS INCH.
PROGRAM 2 WITHOUT MEMBRANE REG. SEA 1
PLOT IS C.G. ACCELERATION VERSUS TIME
FIGURE 8

X-SCALE=5.00E+00 UNITS INCH.
Y-SCALE=2.00E-01 UNITS INCH.
PROGRAM 2 MODEL 5 20KTS REG. SEA 2
PLOT IS C.G. ACCELERATION VERSUS TIME
FIGURE 9

x-scale=5.00E+00 UNITS INCH.
y-scale=1.00E-01 UNITS INCH.
PROGRAM 2 WITHOUT MEMBRANE REG. SEA 2
PLOT IS C.G. ACCELERATION VERSUS TIME
VI. CONCLUSIONS AND RECOMMENDATIONS

A. PROGRAM TIMING

The inclusion of pressure rate and volume rate equations improved the execution time of the L&M program without decreasing the program accuracy. The changes to the program were not difficult to implement. Further effort to improve execution time of the program should be directed toward changes to SUBROUTINE WAVES. In connection with this, a study of the number of required sidewall and bow and stern seal stations should also be conducted. By reducing the number of sidewall stations the execution time should decrease. A study of the losses in accuracy, however, would be necessary.

Another facet for investigation within SUBROUTINE WAVES would be a study to determine if the trigonometric functions could be represented by the in-line programming of the expansion of sine and cosine functions, vice the use of computer library functions. Since some sine and cosine functions are nested in DO LOOPS small savings in time would accumulate. Again, the losses in accuracy to this change would have to be determined.

The elimination of SUBROUTINE DMINV did not produce any noticeable time savings but did save approximately 2 K bytes of core.
The expansion of the number of wave components did not demonstrate any time savings; however the optimum number of wave components required to simulate irregular sea conditions should be determined. A possible course of action would be to generate wave components, calculate the magnitude and frequency and compute the energy spectral density. This energy density spectrum could then be compared to the Pierson and Neumann or Moskowitz model [Ref. 11]. By determining the optimum number of wave components, a savings in CPU execution time could result.

B. ADDITIONAL MODELING

The fact that waves can hit the top of the plenum was discussed in Section V. Whether this effect produces significant forces and moments is unknown. Since the time that the wave is in contact with the plenum and the location (station numbers) is known and since the area of the plenum in contact with the waves could be approximated, (no greater than the difference between wetted stations) additional correction terms could be developed to determine whether significant forces and moments are generated by wave contact. The drag of the wetted plenum top could also be considered.

C. MEMBRANE MODELING

Of the five membrane models developed in this study, Model Five is the most usable. The accuracy of this model must be determined and the model must be validated. Whether
mass and mass flow rate terms could improve this model also needs to be determined. In this regard a detailed study of adiabatic-reversible unsteady flow conditions would be useful. This would mean the establishment of a suitable control volume and the solution of energy-work equations.

The membrane modeling in this study was for a continuous non-permeable membrane. Additional modeling for the case of a hole cut into the membrane should be attempted. The equation for a sharp-edge orifice could be used to determine air loss. The area (size) of the hole can be determined by considering as a first approximation the ratio of surface area of a hemisphere to volume of the hemisphere. As an alternative, the experimental results of the towing tank tests conducted at NSRDC, on the XR-1, using suitable scale factors, could be used to determine empirical correction factors to be applied to the Loads and Motion program.

The occurrences of algebraic loops and summation of feed forward terms as shown in the computational flow diagrams and the failure of the models in which they occur indicate that further modeling of the membrane is required. To prevent model failures, elimination of the effects of the computational loops appears to be necessary.
MEMBRANE SCALE FACTOR

The empirical relationship between pressure and volume for the membrane given in Ref. 9 is for the XR-1 test craft.

\[ V_m = K P_m \]  \hspace{1cm} (A1)

where

\[ K = \frac{40 \text{ in}^3}{\text{Ps}} \]

It was therefore necessary to scale the constant \( K \) for the XR-3 test craft.

In the following development the subscript 1 refers to the XR-1 and subscript 3 refers to XR-3.

Since XR-1 and XR-3 are similar it was assumed that all dimensions scale by the ratio of plenum lengths

\[ \frac{L_3}{L_1} = \lambda \]  \hspace{1cm} (A2)

where

\[ L_3 = 20 \text{ ft.} \]
\[ L_1 = 5.4 \text{ ft.} \]

It was assumed that the ratio of mass to volume was constant

\[ \frac{M_1}{V_1} = \frac{M_3}{V_3} = \rho \]  \hspace{1cm} (A3)
The ratio of craft volume is

\[ \frac{V_3}{V_1} = \lambda^3 \]  \hspace{1cm} (A4)

The ratio of craft pressure is

\[ \frac{P_3}{P_1} = \frac{M_3 g / A_3}{M_1 g / A_1} \]  \hspace{1cm} (A5)

where \( g \) is the gravitational constant.

Substituting (A2) and (A3) into (A5)

\[ \frac{P_3}{P_1} = \frac{V_3 / A_1}{V_1 / A_3} = \frac{\lambda^3}{\lambda^2} = \lambda \]  \hspace{1cm} (A6)

Using (A4) and (A6) in (A1)

\[ K_1 = \frac{V_1}{P_1} \cdot \frac{V_3 / \lambda^3}{P_3 / \lambda} = \frac{K_3}{\lambda^2} \]  \hspace{1cm} (A7)

\[ K_3 = \lambda^2 K_1 \]  \hspace{1cm} (A8)

Converting in.\(^3\) to ft.\(^3\) and using (A2)

\[ K_3 = \left( \frac{20}{5.4} \right)^2 \cdot \frac{40}{1728} = 0.3175 \text{ ft.}^3/\text{psf} \]  \hspace{1cm} (A9)
APPENDIX B

LISTING OF FORTRAN XR-3

L&M COMPUTER PROGRAM

PROGRAM TWO MODEL FIVE
INTEGER ON
COMMON /AIR/ PINF, RHOINF, GAM
COMMON /BMCO/ IMM, IMNX, IMNY, IRMFIL, BTIME, IMT, XMI(10), YMI(7), IX, IY
COMMON /CONST/ PI, RAD, UO
COMMON /ENGINE/ NPS, NPP, THSTS(25), THSTP(25), XP, YP, ZP, STHS, STHP, TIPMA
1(25), TIS(25)
COMMON /EQNC/ NEOS, TOL(20), JOQ
COMMON /FPROP/ FPX, FYP, FZP, FKP, FNP
COMMON /FRONDE/ FNP, FNCRTT
COMMON /PRIME/ STIME, ETIME, DELT, DEMPNT, TPRINT
COMMON /PRMOD/ PROM01, PROMO2, PROMO3, PROMO4, PROMO5, PROMO6, PROMO7
COMMON /PRINT/ ON, IACCEL, IVEL, ITIRAJ, ISIDLW, IBOWSL, ISTNSL, IWAVES, IMAIN
1RUD, IPRP, IACRDO, IFRS
COMMON /ROLL/ PHIMAX, TROLL
COMMON /RUDDR/ NPR, DEMPUD(25), XP, YP, ZP, IRDS, TL, RSPAN, RREA, RASPR, RMAIN
1CLB, RTC, RUCANG, TIR(25)
COMMON /VALOLD/ YOLD(20)
COMMON /VARGL/ VAL(40)
COMMON /WAVE/ ETA(4, 2), AW(20), OMEGA(20), DWH, NVAW, BETA, FXTI, FYMAIN
1WAV, FZWAV, FKMVA, FMDW, ZBAR, PHI BAR, THETAR, TC, COSBET, SINBET, PBMAIN
2BAR
EQUIVALENCE (VAL(2), U), (VAL(3), V), (VAL(4), W), (VAL(5), P), (VAL(6), 1), (VAL(7), R), (VAL(8), PH1), (VAL(9), THETA), (VAL(10), 2), (VAL(11), BMAS), (VAL(21), X), (VAL(22), Y), (VAL(23), PSI), (VAL(24), PB), (VAL(25), VOL)
3 (VAL(12), VOL)
DIMENSION DUMMY(20)
TC = 1.0
ON = 1
PI = 4.0 ATAN(1.)
RAD = 180.0 / PI
WRITE (6, 12)
1 READ (44, 13, END=2) DUMMY
WRITE (6, 14) DUMMY
WRITE (5, 13) DUMMY
GO TO 1
2 REWIND 5
3 CALL INCON (TIME)
IF (IMM.EQ.3) GO TO 11
VAL(13) = VAL(24)
C
DO 4 J=1, 20
4 YOLD(J) = VAL(J+1)
C
GO TO 8
5 CONTINUE
TOL = TIME
PBBAR = PBBAR*(1.0 - DELT/TC) + DELT*(PB-PINF)/TC
MAIN 10
MAIN 20
MAIN 30
MAIN 40
MAIN 50
MAIN 60
MAIN 70
MAIN 80
MAIN 90
MAIN 100
MAIN 110
MAIN 120
MAIN 130
MAIN 140
MAIN 150
MAIN 160
MAIN 170
MAIN 180
MAIN 190
MAIN 200
MAIN 210
MAIN 220
MAIN 230
MAIN 240
MAIN 250
MAIN 260
MAIN 270
MAIN 280
MAIN 290
MAIN 300
MAIN 310
MAIN 320
MAIN 330
MAIN 340
MAIN 350
MAIN 360
MAIN 370
MAIN 380
MAIN 390
MAIN 400
MAIN 410
MAIN 420
MAIN 430
MAIN 440
MAIN 450
MAIN 460
MAIN 470
IF (NWAVE.LE.0) GO TO 6
   ZBAR = (1.0-DELT/TC)*ZBAR+DELT*Z/TC
   PHI = (1.0-DELT/TC)*PHI+DELT*PHI/TC
   THEBAR = (1.0-DELT/TC)*THEBAR+DELT*THEBAR/TC
CALL WAVES (TIME)
6  CALL SIDEWL
   CALL PROP
   CALL RUDDER
   CALL AEROD
   CALL INTGRT (TIME)
   IF (TIME.GT.FTIME) GO TO 10
   IF (FN.GT.FMCRT) GO TO 7
   PRINT 17
   GO TO 10
7  DELOLD = TIME-TOLD
   PSI = PSI+DELODL*R
   X = X+DELOLD*(U*COS(PSI)-V*SIN(PSI))
   Y = Y+DELOLD*(U*SIN(PSI)+V*COS(PSI))
   IF (ABS(TIME-TPRINT).LT.1.0E-6) GO TO 8
   GO TO 5
8  CONTINUE
   IF (ITRAJ.EQ.0) GO TO 9
   DPHI = PHI*RAD
   DPSI = PSI*RAD
   DTHETA = THETA*RAD
   DP = P*RAD
   DQ = Q*RAD
   DR = R*RAD
   VEL = 0.5925*S
   WRITE (6,15) TIME,VEL,V,W,DP,DQ,DR,Z,DPHI,DTHETA,V,Y,DPSI
   BETS = (-V/U)*RAD
   DELRS = RUDANG*RAD
   WRITE (6,16) BETS,DELRS,FXP
9  CONTINUE
   IMM = (IMM+1)/2
   IF (IMM.EQ.1.AND.TIME.GE.BTIME-1.0E-6) IMM=1
   TPRINT = TPRINT+DELPNT
   GO TO 5
10  CALL COLFIL
   IF (IMM.LT.1) GO TO 3
   IF (IMM.NE.1) GO TO 11
   END FILE IBMFIL
   GO TO 3
11  CALL SAM
   GO TO 3
C
12  FORMAT (1H1/35X,22H LISTING OF INPUT DECK//)
MAIN 480
MAIN 490
MAIN 500
MAIN 510
MAIN 520
MAIN 530
MAIN 540
MAIN 550
MAIN 560
MAIN 570
MAIN 580
MAIN 590
MAIN 600
MAIN 610
MAIN 620
MAIN 630
MAIN 640
MAIN 650
MAIN 660
MAIN 670
MAIN 680
MAIN 690
MAIN 700
MAIN 710
MAIN 720
MAIN 730
MAIN 740
MAIN 750
MAIN 760
MAIN 770
MAIN 780
MAIN 790
MAIN 800
MAIN 810
MAIN 820
MAIN 830
MAIN 840
MAIN 850
MAIN 860
MAIN 870
MAIN 880
MAIN 890
MAIN 900
MAIN 910
MAIN 920
MAIN 930
MAIN 940
MAIN 950
SUBROUTINE AEROD
C
INTEGER ON
COMMON /FAERO/ FX, FY, FZ, FK, FM, FN
COMMON /FAIR/ PHOA, XLAERO
COMMON /PROMOD/ PROMO1, PROMO2, PROMO3, PROMO4, PROMO5, PROMO6, PROMO7
COMMON /PRTINT/ ON, IACCEL, IVFL, ITRAJ, ISIDWL, IBDWSL, ISTNSL, IWAYS, IAER
1RUD, ITK, IAERON, IHIRS
COMMON /VARBLE/ VAL(40)
EQUIVALENCE (VAL(2), U), (VAL(3), V), (VAL(4), W), (VAL(5), P), (VAL(6), E)
1, Q), (VAL(7), R), (VAL(8), PHI), (VAL(9), THETA), (VAL(10), Z), (VAL(AER
211), BMASS), (VAL(21), X), (VAL(22), Y), (VAL(23), PSI), (VAL(24), PB)

CA = RHQX*U*U
QAL = QA*XLAERO
BETA = -V/U
BETASQ = BETA*BETA
FX = -(0.90*BETASQ+0.13)*QA
FY = (0.0*BETASQ+0.53*BETA)*QA
FZ = -(2.06*BETASQ+0.39)*QA
FK = -(0.5*BETASQ+0.0*BETA)*QAL
FM = (0.29*BETASQ+0.12)*QAL
FN = (0.0*BETASQ+0.076*BETA)*QAL
IF (IAERO. NE. ON) RETURN
WRITE (6, 1) FX, FY, FZ, FK, FM, FN
RETURN

1 FORMAT (/10X, 23AERON FX, FY, FZ, FK, FM, FN/6E15.4)
END

BLOCK DATA
COMMON /AIR/ Z1(3)
COMMON /GMCO/ Z2(25)
DATA Z6/107*0.0/
DATA Z7/21*0.0/
DATA Z8/6*0.0/
DATA Z9/2*0.0/
DATA Z10/262*0.0/
DATA Z11/7*0.0/
DATA Z12/8*0.0/
DATA Z13/6*0.0/
DATA Z14/2*0.0/
DATA Z15/6*0.0/
DATA Z16/0.0/
DATA Z17/138*0.0/
DATA Z18/62*0.0/
DATA Z19/62*0.0/
DATA Z20/11*0.0/
DATA Z21/0*0.0/
DATA Z22/4*0.0/
DATA Z23/817*0.0/
DATA Z24/36*0.0/
DATA Z25/55*0.0/
DATA Z26/12*0.0/
DATA Z27/4*0.0/
DATA Z28/7*0.0/
DATA Z29/5*0.0/
DATA Z30/12*0.0/
DATA Z31/2*0.0/
DATA Z32/0*0.0/
DATA Z33/2*0.0/
DATA Z34/62*0.0/
DATA Z35/22*0.0/
DATA Z36/20*0.0/
DATA Z37/19*0.0/
DATA Z38/5*0.0/
DATA Z39/2*0.0/
DATA Z40/20*0.0/
DATA Z41/40*0.0/
DATA Z42/100*0.0/
DATA Z43/80*0.0/
DATA Z44/5*0.0/
DATA Z45/7*0.0/
END

SUBROUTINE BOWSL

INTEGER CN
COMMON /AIP/ PINC,RHOINF,GAM
COMMON /CONST/ PI,RAD,UF
COMMON /FORBS/ FX,FY,FZ,FK,FM,FN,QL

BLDA 520
BLDA 530
BLDA 540
BLDA 550
BLDA 560
BLDA 570
BLDA 580
BLDA 590
BLDA 600
BLDA 610
BLDA 620
BLDA 630
BLDA 640
BLDA 650
BLDA 660
BLDA 670
BLDA 680
BLDA 690
BLDA 700
BLDA 710
BLDA 720
BLDA 730
BLDA 740
BLDA 750
BLDA 760
BLDA 770
BLDA 780
BLDA 790
BLDA 800
BLDA 810
BLDA 820
BLDA 830
BLDA 840
BLDA 850
BLDA 860
BLDA 870
BLDA 880
BLDA 890
BLDA 900
BLDA 910
BLDA 920

BWSL 10
BWSL 20
BWSL 30
BWSL 40
BWSL 50
BWSL 60
FK = 0.0
FM = 0.0
FN = 0.0
DELPG = PBS-PB
IF (DELPG.LT.0.0) DELPG=0.0
PBAR = PB-PINF
DELP = PBAR
IF (DELP.LT.0.0) DELP=0.0
ARGO = ELMAXB/CORLEN
ANGO = ARSIN(ARGO)
X1 = XBS+ZBS*THETA-CORLEN*COS(ANGO)
Z1 = 2-ZBS*ABS-THETA-ELMAXB*CSI(THETA)
DPTHFT = (5.5/(1.+(U/25.))*2)*0.0833
IF (CENCAB.GT.1.1875) CENCAB=1.1875
N = NSTA(3)

DO 3 K=1,N
DPTHFT(K) = DPTHFT
ELSKI(K) = (ETA(3,K)-DETABX(K)*(XX(3,K)-X1)-Z1)+YY(3,K)*PHI+XLXPWVB
1*WATSLP
IF (ELSKI(K).GT.HINGHT) ELSKI(K)=HINGHT
IF (((HINGHT-ELSKI(K)+DPTHFT(K)).GE.ELMAXB) DPTHFT(K)=ELMAXB-HINGHT+ELSB
3 K(K)
IF (DPTHFT(K).LT.0.0) DPTHFT(K)=0.0
ELSKID(K) = (ELSKI(K)-DPTHFT(K))*12.0
12.0
DPIN = DPTHFT(K)*12.0
MM = DPIN
MM1 = MM+1
MM2 = MM1+1
DINC = DPIN-MM
GAP(K) = -ELSKI(K)+(HINGHT-ELMAXB)
IF (GAP(K).LT.0.0) GAP(K)=0.0
IF (ELSKID(K).GE.0.) GO TO 2
WETLEN(K) = ELSKI(K)
GO TO 3
2
MM3 = ELSKID(K)
MM4 = MM3+1
MM5 = MM4+1
DLCINC = ELSKID(K)-MM3
BWSL1 = BWSL(MM1,MM4)
BWSL2 = BWSL(MM1,MM5)
BWSL3 = BWSL(MM2,MM4)
BWSL4 = BWSL(MM2,MM5)
BWSL1 = (BWSL2-BWSL1)*DINC+BWSL1
BWSL2 = (BWSL4-BWSL3)*DINC+BWSL3
WETLEN(K) = (((BWSL2-BWSL1)*DINC+BWSL1)*0.08333
3 CONTINUE
C
N = NSTA(3)-1
C
DO 10 J=1,N
WETLAV = (WETLEN(J+1)+WETLEN(J))/0.5
IF (WETLAV.LE.0.001) GO TO 8
DPFTA( J ) = (DPFT( J+1)+DPFT( J))/0.5
ELS KIA = (ELSKI(J+1)+ELSKI(J))/0.5
ELS KDA = (ELSKID(J+1)+ELSKID(J))/0.04166
SEALHT = HINGHT-ELSKDA
DIFF = 2.0*E[NCAB-(SEALHT+0.5)]
IF (DIFF.GT.0.5) DIFF=0.5
ARM1B(J) = X1+WETLAV*0.5
ARM2B(J) = Z5-ELSKIA+DPFTA*0.5
IF (DIFF.GE.0.25) GO TO 4
DFBS(J) = -DELP*DELYBS*WETLAV
GO TO 7
4 FORLEN = XBF-WETLAV
IF (FORLEN.EQ.0.0) GO TO 5
ARGH = (HINGHT-ELSKIA)/FORLEN
IF (ARGH.GT.1.0) ARGH=1.0
ANGW = ARSIN(ARGH)
FORCOS = COS(ANGW)
GO TO 6
5 FORCOS = 0.0
6 DFBS(J) = DELP*DELYBS*WETLAV-DELP*FORLEN*DELYBS*FORCOS*(FORLEN*ORI)
7 ARG = 0.5*RHO*U*U*WETLAV*DELYBS
RESK1 = U*WETLAV/ENU
CDTSK1 = 0.427/(ALOG10(RESK1)-0.407)*2.64
TSKIB(J) = -ARG*CDTSK1
GO TO 9
8 DFBS(J) = 0.0
TSKIB(J) = 0.0
9 CONTINUE
FX = FX+TSKIB(J)
FZ = FZ+DFBS(J)
FK = FK+DFBS(J)*YAVGB(J)
FM = FM-DFBS(J)*ARM1B(J)+TSKIB(J)*ARM2B(J)
FN = FN-LSKIB(J)*YAVGB(J)
ALBS = ALBS+(GAP(J)+GAP(J+1))*DELYBS*0.5
10 CONTINUE
C
ALBS = ALBS+BLEAK
SQFAC = SQRT(2.*ABS(1./PBAR)/RHOINF)
QL = CFBS*ALBS*SQFAC*SIGN(1.,PBAR)
IF (I3CWSL.NE.ON) RETURN
C

BWSL 1030
BWSL 1040
BWSL 1050
BWSL 1060
BWSL 1070
BWSL 1080
BWSL 1090
BWSL 1100
BWSL 1110
BWSL 1120
BWSL 1130
BWSL 1140
BWSL 1150
BWSL 1160
BWSL 1170
BWSL 1180
BWSL 1190
BWSL 1200
BWSL 1210
BWSL 1220
BWSL 1230
BWSL 1240
BWSL 1250
BWSL 1260
BWSL 1270
BWSL 1280
BWSL 1290
BWSL 1300
BWSL 1310
BWSL 1320
BWSL 1330
BWSL 1340
BWSL 1350
BWSL 1360
BWSL 1370
BWSL 1380
BWSL 1390
BWSL 1400
BWSL 1410
BWSL 1420
BWSL 1430
BWSL 1440
BWSL 1450
BWSL 1460
BWSL 1470
BWSL 1480
BWSL 1490
BWSL 1500
WRITE (6,11) GAP, WETLEN, FX, FY, FZ, FK, FM, FN

RETURN

11 FORMAT (/10X, 8HBOW SEAL/26H GAP (FT.) (PORT TO STBD.) /11F10.5/ 28HBWSL FX, FY, FZ, FK, FM, FN)

2N/6E15.4)

END

SUBROUTINE COLFIL

COMMON / AXIS/NXYS(26)
COMMON / COLUMN/ IVERT, ILATRL
COMMON / CURVE/ NCUR(10)
COMMON / CURRENT/ NEQS, TOL(20), JQO
COMMON / GRAF/ NGRAF, NDWR
COMMON / HEADG/TICRD(6)
COMMON / MOD/ MOD01, MOD02, PRM03, PRM04, PRM05, PRM06, PRTM07
COMMON / STEP/ STP2
COMMON / SUM/ ISUM1(8), ISUM2(8)

REAL *8 TICRD
REAL *8 NGEF!
REAL *8 TITLX(12)
REAL *8 LINE(24), *8 PLOTIS, *8 VERSUS/
REAL *8 NAMEX(2), NAMFY(2), TNAME(16)

EQUIVALENCE (TITLE(1), TICRD(1)), (TITLE(2), TICRD(2)), (TITLE(3), TICRD(3))
EQUIVALENCE (TITLE(4), TICRD(4)), (TITLE(5), TICRD(5)), (TITLE(6), TICRD(6))

DIMENSION PVQQ(26), XOUT(900), YOUT(900), AFLE(8)
DIMENSION THST(1), THSTP(1)

EQUIVALENCE (PVQQ(1), TIME), (PVQQ(2), ETA), (PVQQ(3), Z), (PVQQ(4), TICRD)
EQUIVALENCE (PVQQ(5), PHI), (PVQQ(6), VWAC), (PVQQ(7), ACC), (PVQQ(8), FANC)
EQUIVALENCE (PVQQ(9), PHU), (PVQQ(10), BETAS), (PVQQ(11), ACCLAT), (PVQQ(12), TICRD)
EQUIVALENCE (PVQQ(13), TRADUS), (PVQQ(14), VDLP), (PVQQ(15), X), (PVQQ(16), FCL)
EQUIVALENCE (PVQQ(17), QIN), (PVQQ(18), QOUT), (PVQQ(19), GFX), (PVQQ(20), FCL)
EQUIVALENCE (PVQQ(21), THST(1)), (PVQQ(22), THSTP(1)), (PVQQ(23), QDEG), (PVQQ(24), PDEG)
EQUIVALENCE (PVQQ(25), RDEG), (PVQQ(26), DELRS)
IF (JQQ .NE. 2) GO TO 1
WRITE (6,23) STEP2
END FILE 1
REWIND 1
TITLE(7) = LINE2(1)
TITLE(10) = LINE2(2)
IF (NGRAF .EQ. 0) GO TO 7
J = 1
NGF = NGRAF
INDEX = NGRAF*2
C
DO 6 I=1,INDEX,2
INDX = NXYS(I)
INDY = NXYS(I+1)
IC = 0
2 READ (1,END = 8) TIME,ETA,Z,THETA,PB,BOW,ACC,ACC,FANPWR,PHI,BETAS,ACEL
ICLAT,UC,TRADIUS,VOLP,X,Y,QIN,QOUT,GFXXX,FPXPAV,THSTS(I),THSTP(I),QDCFL
2FG,PDEG,RDEG,DELRS
IF (IC.GE.900) GO TO 3
IC = IC+1
XOUT(IQ) = PVQQ(INDX)
YOUT(IQ) = PVQQ(INDY)
GO TO 2
3 REWIND 1
INX = INDX*2
INY = INDY*2
NAMEX(1) = NAMES(INX-1)
NAMEX(2) = NAMES(INX)
NAMEY(1) = NAMES(INY-1)
NAMEY(2) = NAMES(INY)
IF (NDRW .EQ. 1) GO TO 4
C
CALL PLOT P(XOUT,YOUT,-IQ,0)
C
WRITE (6,24) NAMEX,NAMEY
GO TO 6
4 TITLE(8) = NAMEY(1)
TITLE(9) = NAMEY(2)
TITLE(11) = NAMEX(1)
TITLE(12) = NAMEX(2)
NCUR = NCURV(J)
IF (NCUR .NE. 0) GO TO 5
LABEL = LABI(1)
C
CALL DRAW (IQ,XOUT,YOUT,NCUR,0,LAB1,TITLE,0,0,0,0,0,8,8,0,LAST)
C
J = J+1
GO TO 6
5 WRITE (2) IQ, NCUR, NGF, LABEL, (TITLE(K), K=1, 12), (XOUT(L), YOUT(L)), L=1 CF1 870
   1, (C)
   NGF = NGF - 1
6 CONTINUE
C
7 IF ((NGF.EQ.0).AND.(NCUR.EQ.3)) GO TO 10
8 IF (IVERT.NE.1) GO TO 16
   WRITE (6, 25)
   K = 0
9 CONTINUE
C
10 DO 8 I=1, 8
   IF (ISUM1(I).NE.0) K = K + 1
8 CONTINUE
C
11 NUM1 = K
12 IF (K.EQ.0) GO TO 16
13 N = K * 2
14 J = 1
15 CONTINUE
C
16 DO 9 I = 1, NUM1
17 INDEX = ISUM1(I) * 2
18 INAME(J) = NAMES(INDEX - 1)
19 INAME(J + 1) = NAMES(INDEX)
20 J = J + 2
9 CONTINUE
C
21 WRITE (6, 26) (INAME(I), I=1, N)
22 GO TO 14
23 END FILE 2
24 REWIND 2
25 NGF = NGMF
26 J = 1
27 READ (2, END = 38) IQ, NCUR, NGMF, LABEL, (TITLE(K), K=1, 12), (XOUT(L), YOUT(L)), L=1 CF1 1190
   10, 1, IQ)
   IF ((NGF.EQ.NGMF).AND.(NCUR.EQ.J)) GO TO 12
28 GO TO 11
29 LABEL = LAB(J+1)
30 CALL DRAW (IQ, XOUT, YOUT, NCUR, 0, LABEL, TITLE, 0, 0, 0, 0, 8, 8, 0, LAST) CF1 1250
31 CONTINUE
C
32 REWIND 2
33 J = J + 1
34 IF (J.EQ.4) GO TO 13
35 GO TO 11
36 NGF = NGF - 1
37 IF (NGF.EQ.0) GO TO 7
38 J = 1
39 GO TO 11
C 14 READ (1,END = 13) (PVQQ(I),I=1,26)
    DO 15 I=1,NUM1
       J = ISUMI(I)
    15 AFILE(I) = PVQQ(J)
C WRITE (6,27) (AFILE(I),I=1,NUM1)
C GO TO 14
C REMIND 1
    16 IF (ILATRL.NE.1) GO TO 22
    WRITE (6,28)
    K = 0
C DO 17 I=1,8
    IF (ISUM2(I).NE.0) K = K + 1
    17 CONTINUE
C NUM2 = K
    IF (K.EQ.0) GO TO 21
    N = K + 2
    J = 1
C DO 18 I=1,NUM2
    IDFX = ISUM2(I)*2
    INAME(J) = NAMES(IDEX-1)
    INAME(J+1) = NAMES(IDEX)
    J = J + 2
    18 CONTINUE
C WRITE (6,26) (INAME(I),I=1,N)
C READ (1,END = 16) (PVQQ(I),I=1,26)
C DO 20 I=1,NUM2
       J = ISUM2(I)
    20 AFILE(I) = PVQQ(J)
C WRITE (6,27) (AFILE(I),I=1,NUM2)
C GO TO 19
    21 REMIND 1
    22 RETURN
C 23 FORMAT ('0',4X,'THIS RUN USED VARIABLE STEP SIZE','/','0',4X,'THE MINIMUM STEPSIZE RECORDED DURING THE RUN WAS','2X,F30.5)
C 24 FORMAT ('0',20X,'2A8','IS THE INDEPENDENT VARIABLE AND ','2A8',' THE INDEPENDENT VARIABLE')
C 25 FORMAT ('0',50X,'***SUMMARY ONE***','/','0')
C 26 FORMAT ('0',16A8)
C 27 FORMAT ('0',8(3X,F10.2,3X))
SUBROUTINE FAN
C
INTEGER ON
COMMON /AIRMAP/ QIN, QBFAN(25), QMFAN(25), QSFAN(25), ENBFAN, ENMFAN, ENFAN
1 SFAN, BRPF, EMRFM, SRPM, NTPS, NPTS, NPTSS, PBDFAN(25), PMFAN(25), PSFAN(25), NS
25, TMEB(25), DELB(25), NB, TEMS(25), DELS(25), NS
 COMMON /PROMOD/ PROM1, PROM2, PROM6, PROM7, PROM05, PROM06, PROM07
 COMMON /PRINT/ ON, IAACC, IVEL, ITRA, ISTDL, IBOWSL, ISTNSL, IWAVE, IFAN
 COMMON /PROP, IAERO, IRS/ FAN
 COMMON /SOFTS/ XBF, PBS, SINRS, COSBS, XBS, ZBS, DELYBS, DPBS, EMXMB, YAVFAN
1 GB(10)
 COMMON /SOFTSS/ XLF, FSS, SINTH, COSTH, XSS, ZSS, DELYSS, DPSS, EMXSS, YAVFAN
1 GS(10)
 COMMON /VARBLE/ VAL(40)
 DIMENSION QB(1), QM(1), QF(1), PBOW(1), PM(1), PS(1), HP(8)
 EQUIVALENCE (VAL(1), TIME), (VAL(2), I), (VAL(3), V), (VAL(4), W), (VAL(5), 3)
 1 (VAL(6), T), (VAL(7), R), (VAL(8), PH), (VAL(9), THETA), (VAL(10), Z)
 2 (VAL(11), BM), (VAL(12), X), (VAL(22), Y), (VAL(23), PSI)
 3, (VAL(24), PB)
 EQUIVALENCE (VAL(18), FANPRW)
 EQUIVALENCE (QBFAN(1), QB(1)), (QMFAN(1), QM(1)), (QSFAN(1), QF(1)),
1 (PBDFAN(1), PBOW(1)), (PMFAN(1), PM(1)), (PSFAN(1), PS(1))
 CATA HP/2.9, 2.3, 1.95, 1.77, 1.85, 2.05, 1.93, 1.62/
 C
 BRAT = 8000/BRPM
 EBRAT = 8000/EMRFM
 SRAT = 8000/SRPM
 TL = VAL(1)
 IF (NB.EQ.0.0) GO TO 1
 DPBS = FG1(TL, NB, TMEB, DELB, ILB)
 PBS = PB+DPBS
 1 IF (NS.EQ.0.0) GO TO 2
 DPSS = FG1(TL, NS, TEMS, DELS, ILS)
 PSS = PB+DPSS
 CONTINUE
 PB1 = PBS-PINF
 PB2 = PB-PINF
 PB3 = PSS-PINF
 PBARB = PB1*BRAT*BRAT
 PBARM = PB2*EMRFM*EMRFM
 PBARS = PB3*SRAT*SRAT
 QBOW = ENBFAN*FG1(PBAR, NTPS, PBOW, QB, BRAT)
 QM4IN = ENMFAN*FG1(PBAR, NPTS, PM, QB, IM)*EMRFM
 QSTN = ENSFAN*FG1(PBAR, NPTSS, PS, QB, IS)/SRAT

28 FORMAT ('0',50X,'***SUMMARY TWO***',/,'0')

CFL 1830
CFL 1840
QIN = QBOW+QMAIN+QSTN
MB1 = (QBOW/ENBFAN+5.0)/5.0
MB2 = MB1+1
MB3 = MB2+1
BIN = ((QBOW/ENBFAN+5.0)-MB1*5.0)/5.0
BFANHP = ((HP(MB3)-HP(MB2))*BIN+HP(MB2))*FNBFAN*(1./BRAT)**3
MS1 = (QSTN/ENSFAN+5.0)/5.0
MS2 = MS1+1
MS3 = MS2+1
STINC = ((QSTN/ENSFAN+5.0)-MS1*5.0)/5.0
SFANHP = ((HP(MS3)-HP(MS2))*STINC+HP(MS2))*ENSFAN*(1./SRAT)**3
MM1 = (QMAIN/ENMFAN+5.0)/5.0
MM2 = MM1+1
MM3 = MM2+1
PLINC = ((QMAIN/ENMFAN+5.0)-MM1*5.0)/5.0
PFANHP = ((HP(MM3)-HP(MM2))*PLINC+HP(MM2))*ENMFAN*(1./EMRAT)**3
RELPWR = PFANHP+BFANHP+SFANHP
FANPWR = (QBOW*PB1+QMAIN*PB2+QSTN*PB3)/550.
FANEFF = FANPWR/RELPWR
IF (IRHS.NE.ON) RETURN
WRITE (6,3) QBOW,QMAIN,QSTN,PBARB,PBARM,PBARS,RELPWR,FANPWR,FANEFF
RETURN
3 FORMAT (/4H FAN/32H Q - BOW,MAIN,STERN (CU FT /SEC)3F12.1/28H DELP,FAN 710
1P - BOW,MAIN,STERN (PSF)3F11.2/60H ACTUAL FAN POWER REQUIRED(HP),
21DEAL FAN POWER, EFFICIENCY 3F12.4)
END

FUNCTION FG1 (X,N,XT,YT,IX)
DIMENSION XT(1), YT(1)
IF (IX.LT.1) IX = 1
IF (IX.GT.N-1) IX = N-1
I = SIGN(1.,0.-XT(IX))
1 IF (IX.LT.1 .OR. IX.GE.N) GO TO 3
IF (XT(IX).GT.0. .OR. XT(IX).LT(XT(IX)+1)) GO TO 2
C = (XT(IX))/XT(IX+1)-XT(IX))
GO TO 4
2 IX = IX+1
GO TO 1
3 C = IX/N
IX = IX-1
4 FG1 = YT(IX)+C*(YT(IX+1)-YT(IX))
RETURN
END
1,2), ISYS

PROGRAM CONTROL PARAMETERS

4 CONTINUE
GO TO (5,8,9,10,11,12), IOPT

5 CONTINUE
STIME = TEMP(1)
ETIME = TEMP(2)
DELO = TEMP(3)
DELPNT = TEMP(4)
TPRINO = TEMP(5)
IF (TPRINO.LT.STIME+DELPNT) TPRINO=STIME+DELPNT
IF (DELO.GT.DEKPNT) DELO=DELPNT
IF (DELO.EQ.0.0) GO TO 13
GO TO 3

6 READ (5,101) NCURV
GO TO 3

7 READ (5,102) ISUM1
READ (5,102) !SUM2
GO TO 3

8 READ (5,108) IACCEL,IVEL,ITRAJ,ISIDWL,IBOWS,ISTNSL,IWAVES,IRUD,IP,
INOC,IAERO,IRHS
GO TO 3

9 READ (5,111) NEOS,JQQ,(TOL(J),J=1,NEOS)
GO TO 3

10 READ (5,108) IVERT,ILATRL,NVD,NVI,NLD,NLI
GO TO 3

11 CONTINUE
I3DOF = TEMP(1)
ISRGE = TEMP(2)
ITRIM = TEMP(3)
IDIA = TEMP(4)
GO TO 3

12 CONTINUE
PRCM01 = TEMP(1)
PRCM02 = TEMP(2)
PRM03 = TEMP(3)
PRM04 = TEMP(4)
PRM05 = TEMP(5)
PRM06 = TEMP(6)
PRM07 = TEMP(7)
GO TO 3

13 WRITE (6,110)
STOP

C MASS DISTRIBUTION

INCN 960
INCN 970
INCN 980
INCN 1000
INCN 1010
INCN 1020
INCN 1030
INCN 1040
INCN 1050
INCN 1060
INCN 1070
INCN 1080
INCN 1090
INCN 1100
INCN 1110
INCN 1120
INCN 1130
INCN 1140
INCN 1150
INCN 1160
INCN 1170
INCN 1180
INCN 1190
INCN 1200
INCN 1210
INCN 1220
INCN 1230
INCN 1240
INCN 1250
INCN 1260
INCN 1270
INCN 1280
INCN 1290
INCN 1300
INCN 1310
INCN 1320
INCN 1330
INCN 1340
INCN 1350
INCN 1360
INCN 1370
INCN 1380
INCN 1390
INCN 1400
INCN 1410
INCN 1420
INCN 1430
14 G = 32.17
RHO = 1.99
HRHO = RHO*.5
GO TO (15,17,22), INPT
15 IMM = 0
WEIGHT = TEMP(1)
AM = WEIGHT/G
XS = TEMP(2)
ZS = TEMP(3)
AIXX = TEMP(4)
AIYY = TEMP(5)
AIZZ = TEMP(6)
AIXZ = TEMP(7)

C
INEFFIA MATRIX OPERATIONS

16 AMASSI = 1.0/AM
D = 1.0/(AIXX*AIYY-AIXZ*AIXZ)
CIXX = AIXX*D
CIXZ = AIXZ*D
CIZZ = AIZZ*D
AIYYI = 1.0/AIYY
GO TO 3

C
READ WEIGHT DISTRIBUTION - ASSUME TRANSVERSE (PORT/STBD) SYMMETRY
X INPUT DIST. FWD. OF (SIDEWALL) TRANSOM
Y INPUT DIST. TO STARBOARD
Z INPUT DIST. UP FROM KEEL LINE

17 I = 1
18 READ (5,109) AMI(I),XI(I),YI(I),XI(I)
IF (AMI(I).LT.0.0) GO TO 19
I = I+1
IF (I.GT.201) GO TO 97
GO TO 18
19 NMASS = I-1
SUM = 0.0
SUX = 0.0
SUZ = 0.0

C
DO 20 I=1,NMASS
AMI(I) = AMI(I)/G
SUM = SUM+AMI(I)
SUX = SUX+AMI(I)*XI(I)
SUZ = SUZ+AMI(I)*ZI(I)
20 C
AM = SUM*.21
WEIGHT = AM*G
ZS = SUZ/SUM
XS = SLX/SUM
SUM = 0.0
SUX = 0.0
SUY = 0.0
SUZ = 0.0

C
DO 21 I=1,NMASS
XI(I) = XI(I)-XS
ZI(I) = -ZI(I)+ZS
AMK = AMI(I)*2.0
SUX = SUX+AMK*XI(I)*XI(I)
SUY = SUY+AMK*XI(I)*XI(I)
SUZ = SUZ+AMK*ZI(I)*ZI(I)
21 SUM = SUM+AMK*XI(I)*ZI(I)

C
AIXX = SUY+SUZ
AIYY = SUX+SUZ
AIZ = SUX+SUY
AIZ = SUM
GO TO 16

22 GO TO 3

C
XX AND YY TABLES

C
23 CONTINUE
NSTA(1) = TEMP(1)
NSTA(2) = TEMP(2)
NSTA(3) = TEMP(3)
NSTA(4) = TEMP(4)
XTOT = TEMP(5)
GO TO 3

C
SIDEWALL (INCLUDING APPENDAGES)

C
24 CONTINUE
GO TO (25,26), IOPT

25 YSW = TEMP(1)
XLSW = TEMP(2)
CFSW = TEMP(3)
CDSW = TEMP(4)
AVBMSW = TEMP(5)
READ (10) ZZZ
REWIND 10
GO TO 3

C
BLOCK 4 OPTION 2 REMOVED.  NO APPENDAGES
26 CONTINUE
   GO TO 3

27 CONTINUE
   XSSI = TEMP(1)
   ZSSI = TEMP(2)
   ALEAK = TEMP(3)
   CFSS = TEMP(4)
   ELMAXS = TEMP(5)
   DPSS = TEMP(6)
   XL = TEMP(7)
   ARGA = ELMAXS/XL
   COSTH = ARGA
   THSSI = ARCOS(ARGA)
   SINTH = SIN(THSSI)
   GO TO 3

28 CONTINUE
   XBSI = TEMP(1)
   CFBS = TEMP(2)
   DPBS = TEMP(3)
   ZBSI = TEMP(4)
   ELMAXB = TEMP(5)
   XBF = TEMP(6)
   ALEAK = TEMP(7)
   GO TO 3

30 CENCAB = TEMP(1)
   GO TO 3

31 CONTINUE
   GO TO (32,33), IOPT

32 CONTINUE
   XLBW = TEMP(1)
   XBBW = TEMP(2)
   XPWV = TEMP(3)
   WIDTH = TEMP(4)
   XL = TEMP(5)
   XCPD = TEMP(6)
   BUBHGT = TEMP(7)
   XLXPWV = XLBW-XPWV
   PWVHT = (XPWV*XPWV-XLXPWV*XLXPWV)*.5/XL

INCN 2400
INCN 2410
INCN 2420
INCN 2430
INCN 2440
INCN 2450
INCN 2460
INCN 2470
INCN 2480
INCN 2490
INCN 2500
INCN 2510
INCN 2520
INCN 2530
INCN 2540
INCN 2550
INCN 2560
INCN 2570
INCN 2580
INCN 2590
INCN 2600
INCN 2610
INCN 2620
INCN 2630
INCN 2640
INCN 2650
INCN 2660
INCN 2670
INCN 2680
INCN 2690
INCN 2700
INCN 2710
INCN 2720
INCN 2730
INCN 2740
INCN 2750
INCN 2760
INCN 2770
INCN 2780
INCN 2790
INCN 2800
INCN 2810
INCN 2820
INCN 2830
INCN 2840
INCN 2850
INCN 2860
INCN 2870
XPWVXS = XPWV-XS
ABW = XRBW*XLBW
AB = WIDTH*XL
VOLH = (ABW+AB)*BUBHGT*.5
GO TO 3
33 CONTINUE
FCRT = TEMP(1)
GO TO 3
C
C PROPULSION
C
34 CONTINUE
GO TO (35,36), IOPT
35 CONTINUE
XPO = TEMP(1)
YPO = TEMP(2)
ZPO = TEMP(3)
GO TO 3
C
C BLOCK 8 OPTION 2 REMOVED. ENGINE OUT INPUT IN BLOCK 16
C
36 CONTINUE
GO TO 3
C
C RUDDER
C
37 CONTINUE
GO TO (38,39,40), IOPT
38 XRO = TEMP(1)
YR = TEMP(2)
ZR = TEMP(3)
RSPAN = TEMP(4)
RASPR = TEMP(5)
AREA = TEMP(6)
RCLB = 2.*PI*RASPR/(RASPR+2.)
RTC = TEMP(7)
GO TO 3
C
C 910 NOT USED
C
39 CONTINUE
GO TO 3
40 CONTINUE
GO TO 3
C
C AERODYNAMICS
C
41 CONTINUE
XLAERO = TEMP(1)
BEAM = TEMP(2)
RHOA = .5*RHOINF*XLAERO*BEAM
GO TO 3
C
WAVES
C
42 CONTINUE
WAVSW = ICPT
IF (IWAVSW.GT.4) GO TO 97
NWave = TEMP(1)
IF (NWave.EQ.0) GO TO 3
IF (NWave.GT.20) GO TO 97
BETAD = TEMP(2)
BETA = BETAD/RAD
COSBET = COS(BETA)
SINBET = SIN(BETA)
TC = 1.0
GO TO (43,45,47,47), IWAVSW
C
43 DO 44 I=1,NWave
44 READ (5,116) OMFGA(I),AW(I)
C
GO TO 3
C
45 DC 46 I=1,NWave
46 READ (5,116) WAVLEN(I),AW(I)
C
GO TO 3
47 SHTWV = TEMP(3)
G1G = 32.17
G2 = G1G*G1G
G4 = G2*G2
GO TO (3,3,48,49), IWAVSW
C
48 CONTINUE
PERL = TEMP(4)
PERH = TEMP(5)
WNN = (2.0*3.141592)*(1.0/PERH)
WXX = (2.0*3.141592)*(1.0/PERL)
GO TO 50
C
49 CONTINUE
WNN = TEMP(4)
WXX = TEMP(5)
C
50 CONTINUE
UUU = SQRT(SHTWV*54.0)*1.6878
UU4 = UUU**4
C = (WXX/WNN)**(1./NWave)
WHPO = WNN
DO 51 I=1,NWAVE
WWPN = WWPO*CSC
WW = (WWPN+WWPO)*.5
DDW = WWPN-WWPO
WWPO = WWPN
WW4 = WW*WW*WW*WW
WW5 = WW*WW4
SS = 0.0081*G2/(EXP(0.74*G4/(WW4*UU4)))*WW5
CMEGA(I) = WW
AN(I) = SQRT(2.*SS*DDW)
51 CONTINUE
C
GO TO 3
C
INITIAL CONDITIONS
C
52 CONTINUE
UC = TFMP(1)
THTO = TEMP(2)
DSO = TEMP(3)
DELPI = TEMP(4)
DFPHI = TEMP(5)
GO TO 3
53 CONTINUE
C
INPUT COMPLETED. 1) PRINT ALL INPUT
WRITE (6,123) TITLC
WRITE (6,124) STIME,FTIME,DEL0,TPRINO,DELPNT
WRITE (6,125) IACCEL,IVEL,ITFAJ,ISIDWL,IROWSL,ISTNSL,IWAVES,IRUD,IPROP,IAERO,IRHS
WRITE (6,135) ISDOF,ISRGE,ITPIM,IDIA
WRITE (6,142) PROMO1,PROMO2,PROMO3,PROMO4,PROMO5,PROMO6,PROMO7
WRITE (6,126) NEQS,( TDL(J),J=1,NEQS)
WRITE (6,113) WEIGHT,XS,MS,AIXX,AIYY,AIZZ,AIXZ
WRITE (6,112) A1,AIMAX
WRITE (6,135) NSTA
WRITE (6,114) YSW,LSW,CSW,DSW,VANGLE,VSPAN,VCHORD,VXO,VY,VZ0,AV
WRITE (6,115) NAL,DAL,SAL,NDS,DDS,SDS,SNS,NTN,DTH,STH,NBB,DBB,SSB
IF (IMM,GT,0) WRITE (6,121) (XMO(J),J=1,IMMX)
WRITE (6,119) IMM,IMNX,IMNY,IMMFIL,BTIME,IMT
IF (IMM,GT,0) WRITE (6,122) (YMI(J),J=1,IMNY)
WRITE (6,128) XLBW,XBBX
WRITE (6,129) XL WIDTH,XCPO,VOLNOM,OBHGT
WRITE (6,134) DELPI
WRITE (6,127) FNCRT,XLTOT
WRITE (6,141) ENBFAN, BRPM, ENMFAN, EMPM, ENSFAN, SRPM
WRITE (6,131) XR0, YR, ZR0, RONO, PMAX0, RRAT0, REVO, DLRDO, RSPAN, PASPR
WRITE (6,130) XPSI, YPSI, ZPSI, ALPSI, CFSS, ELMAXB
WRITE (6,138) XSSI, ZSSI, ALEAK, CFSS, ELMAXS, DPSS, XLF
WRITE (6,132) U0, THET0, DSO

AND 2) INITIALIZE VARIABLES FOR CALC.

DO 54 I=1,40
  54 VAL(I) = 0.0

U = U0*1.6889
XSS = -(XS-XSSI)
ZSS = ZS-ZSSI
THETA = THETO/RAD
PHI = DPHI/RAD
THEQ = THETA
DS = DSO*0.0833
Z = -ZS+DS
ZEQUI = Z
PHIMAX = 0.
TROLL = 0.
IRDS = 0.
TL = 0.0

WAVE PARAMETERS TABLE
IF (NWave.EQ.0) GO TO 61
AMPTC = 1.30287
GO TO (55,57), IWAVE

55 DO 56 I=1,NWave
  56 WAVLEN(I) = 2.*PI/G/(OMEGA(I)*OMEGA(I))
GO TO 59

57 DO 58 I=1,NWave
  58 OMEGA(I) = SQRT(2.*PI*G/WAVLEN(I))
CONTINUE

CALCULATE INITIAL FREQUENCIES OF ENCOUNTER
DO 60 I=1,NWAVE
WAVSLP(I) = 360.0*AW(I)/WAVLEN(I)
CMEGA(I) = 2.*PI*(SORT(G*WAVLEN(I)/(2.*PI))-U*COSBET)/WAVLEN(I)
ENCPER(I) = 2.0*PI/OMEGA(I)
60 CONTINUE

C WRITE (6,117) NWAVE,BETAD,(OMEGA(I),OMEGA(I),WAVLEN(I),AW(I),WAVSLP(I),CMEGA(I),ENCPER(I),I=1,NWAVE)
GO TO 62
61 WRITE (6,118)
62 CONTINUE

C DO 63 I=1,4

C DO 63 N=1,11

63 ETA(I,N) = 0.0

C DVLW = 0.0
FXWAV = 0.0
FYWAV = 0.0
FZWAV = 0.0
FKWAV = 0.0
FMWAV = 0.0
FNWAV = 0.0
ZBAR = Z
PHIBAR = PHI
THEBAR = THETA
TIME = STIME
DELT = DELC
TPRINT = TPRINO-Delpnt
PWVCON = 4.*WEIGHT/(RHO*G*XLBW)
FNCON = SQRT(XLBW*G)
VX = VXO-XS
VZ = ZS-VZO
XP = XPO-XS
XR = XRO-XS
YP = YPO
ZP = ZS-ZPO
ZR = ZS-ZFO
IF (IMN.EQ.0) GO TO 65

C DO 64 J=1,IMNX

64 XM(J) = XMO(J)-XS

C 65 CONTINUE
XCP = XCP0-XS
ZCP = ZS-BUBHGT
XBS = XBSI-XS
N = NSTA(3)
ZBS = ZS-ZBSI
C
DO 66 J=1,N
DELYBS = XBBW/(N-1)
XX(3,J) = XBS-XSSI
YY(3,J) = -0.5*XBBW+(J-1)*DELYBS
66 CONTINUE
C
N = N-1
C
DO 67 J=1,N
YAVGB(J) = (YY(3,J+1)+YY(3,J))*5
67 CONTINUE
C
N = NSTA(4)
DELYSS = XBBW/(N-1)
C
DO 68 J=1,N
XX(4,J) = XS
YY(4,J) = .5*XBBW+(J-1)*DELYSS
68 CONTINUE
C
N = N-1
C
DO 69 J=1,N
YAVGS(J) = (YY(4,J+1)+YY(4,J))*5
69 CONTINUE
C
XBBW = XLTOT-XS
N = NSTA(1)
DELX = XBSI/(N-1)
C
DO 70 J=1,2
C
DO 70 I=1,N
XX(J,I) = (I-1)*DELX-XS
70 YY(J,I) = YSW(2*J-3)
C
WRITE (6,103) (XX(J,N),N=1,11),(YY(J,N),N=1,11),J=1,4)
N = NSTA(1)-1
C
DO 71 J=1,N
71 XAVG(J) = DELX*(2*I-1)*5-XS
C
CALL WAVES (TIME)
INITIALIZE BUBBLE PRESSURE, ABSOLUTE (PSF)

PB = PINF + CELPI
PPBAR = DELPI
PBAR = DELPI
FSS = PB + DPSS
PBS = PB + DPBS

AB = ABW - (ABW - AB) * (ZS + Z/BUHGT)
C0 = 0.37 / (U/FNCON) ** 1.5655981
WATSLP = PPBAR * CF * PWVCON/WEIGHT

IF (IDIA.EQ.1) GO TO 72
VOL = VOLNUM - 5 * (AB + ABW) * (ZS + Z) - D VOLW + 5 * WATSLP * XL * AB
GO TO 73

72 VOL = VOLNUM - 5 * (AB + ABW) * (ZS + Z) - D VOLW + PPBAR * 3175 * 5 * WATSLP * XL * AB
CONTINUE

BM ASS = (PB/PINF) ** (1. / GAM) * VOL * RHOPINF
WRITE (6, 137)
DVWDOT = 0.0
RET U RN

RUN TERMINATOR

WRITE (6, 106)
STOP

BENDING MOMENT

GO TO (76, 77, 79, 81), IOPT

75 IMM = TEMP(1)
76 IMM = TEMP(1)
IF (IMM.GT.3) GO TO 97
IMNX = TEMP(2)
IF (IMNX.GT.10) GO TO 97
IMNY = TEMP(3)
IF (IMNY.GT.7) GO TO 97
IMMFIL = TEMP(4)
BTIME = TEMP(5)
IM = TEMP(6)
GO TO 3

77 GO 78 J = 1
78 XMO(J) = TEMP(J)

IF (IMNX.LT.7) GO TO 3
READ 120, (XMO(J), J = 8, IMNX)
GO TO 3

79 DO 80 J = 1, IMNY
80 YMI(J) = TEMP(J)

INCN5760
INCN5770
INCN5780
INCN5790
INCN5800
INCN5810
INCN5820
INCN5830
INCN5840
INCN5850
INCN5860
INCN5870
INCN5880
INCN5890
INCN5900
INCN5910
INCN5920
INCN5930
INCN5940
INCN5950
INCN5960
INCN5970
INCN5980
INCN5990
INCN6000
INCN6010
INCN6020
INCN6030
INCN6040
INCN6050
INCN6060
INCN6070
INCN6080
INCN6090
INCN6100
INCN6110
INCN6120
INCN6130
INCN6140
INCN6150
INCN6160
INCN6170
INCN6180
INCN6190
INCN6200
INCN6210
INCN6220
INCN6230
GO TO 3
81 CONTINUE
GO TO 3
82 CONTINUE
GO TO (83,85,87), IOPT
83 CONTINUE

VALUES INPUT FOR STBD SCREW
THST1 = TEMP(1)
NPS = TEMP(2)
STHS = TEMP(3)
IF (NPS.EQ.0.0) GO TO 84
READ (5,104) (TJS(J),J=1,NPS)
READ (5,104) (THSTS(J),J=1,NPS)
GO TO 3
84 THSTS(1) = THST1
85 CONTINUE

VALUES INPUT FOR PORT SCREW
THST2 = TEMP(1)
NPP = TEMP(2)
STHP = TEMP(3)
IF (NPP.EQ.0.0) GO TO 86
READ (5,104) (TJP(J),J=1,NPP)
READ (5,104) (THSTP(J),J=1,NPP)
GO TO 3
86 THSTP(1) = THST2
87 CONTINUE

VALUES INPUT FOR RUDDER
DELR = TEMP(1)
NPR = TEMP(2)
IF (NPR.EQ.0.0) GO TO 88
READ (5,104) (TJR(J),J=1,NPR)
READ (5,104) (DELRU(J),J=1,NPR)
GO TO 3
88 DELRUD(1) = DELR
GO TO 3
89 GO TO (90,91), IOPT
90 NB = TEMP(1)
READ (5,104) (TMEB(I),I=1,NB)
READ (5,104) (DELB(I),I=1,NB)
GO TO 3
91 NS = TEMP(1)
READ (5,104) (TMES(I),I=1,NS)
READ (5,104) (DETS(I),I=1,NS)
GO TO 3

C TITLE CARD (ALL 80 COLUMNS)

92 READ (5,136) TITLC
GO TO 3

C FAN MAPS

93 CONTINUE
GO TO (94,95,96), IOPT

94 CONTINUE
ENBFAN = TEMP(1)
BRPM = TEMP(2)
NPTSB = TEMP(3)
READIN = TEMP(4)
IF (READIN.EQ.0.0) GO TO 3
READ (5,104) (PBFAN(J),J=1,NPTSB)
READ (5,104) (QBFAN(J),J=1,NPTSB)
GO TO 3

95 CONTINUE
ENMFAN = TEMP(1)
EMRPM = TEMP(2)
NPTSM = TEMP(3)
READIN = TEMP(4)
IF (READIN.EQ.0.0) GO TO 3
READ (5,104) (PMFAN(J),J=1,NPTSM)
READ (5,104) (QMFAN(J),J=1,NPTSM)
GO TO 3

96 CONTINUE
ENSFAN = TEMP(1)
SRPH = TEMP(2)
NPTS = TEMP(3)
READIN = TEMP(4)
IF (READIN.EQ.0.0) GO TO 3
READ (5,104) (PSFAN(J),J=1,NPTS)
READ (5,104) (QSFAN(J),J=1,NPTS)
GO TO 3

C ERROR IN INPUT

97 CONTINUE
WRITE (6,105) ISYS
STOP

C

C
98 FORMAT (2(2))  INCN7200
99 FORMAT (26(12))  INCN7210
100 FORMAT (6A8)  INCN7220
101 FORMAT (10(1))  INCN7230
102 FORMAT (8(12))  INCN7240
103 FORMAT (//7H XX AND YY ARRAYS/14H PORT SIDEWALL/2(11F10.2)/,15H SINCN7250
1TBD. SIDEWALL/2(11F10.2)/,9H BOW SEAL/2(11F10.2)/,11H STERN SEAL/2INCN7260
2(11F10.2)/))  INCN7270
104 FORMAT (8F10.0)  INCN7280
105 FORMAT (34H INPUT ERROR - -- STOP -- ISYS=,I3)  INCN7290
106 FORMAT (1H1,20(1/,50X,19H COMPLETED ALL RUNS)  INCN7300
107 FORMAT (13,12,7F10.0)  INCN7310
108 FORMAT (16(5))  INCN7320
109 FORMAT (5F10.0)  INCN7330
110 FORMAT (//10X,65H ERROR IN INPUT --- DELT AND/OR DELPNT EQUALS ZERO)  INCN7340
10 -- -- JOB ABORTED)  INCN7350
111 FORMAT (212/(8F10.0))  INCN7360
112 FORMAT (22H INERTIA MATRIX, AIMAX6E15.4/(22X,6E15.4))  INCN7370
113 FORMAT (30H WEIGHT, C.G. INERTIA MOMENTS7F12.3)  INCN7380
114 FORMAT (15H SIDEWALL INPUT12(F.3,1X))  INCN7390
115 FORMAT (26H SIDEWALL TABLE PARAMETERS4(I4,F7.3,F7.3))  INCN7400
116 FORMAT (2F10.0)  INCN7410
117 FORMAT (/12HOMO OF WAVES12,10H BETA(DEG)F5.0/15H OMEGA(RAD/SEC),5XINCN7420
1,15H OMEGA(RAD/SEC)5X,16H WAVE LENGTH(FT)5X,14H AMPLITUDE(FT)5X,1INCN7430
26H MAX SLOPE (DEG)5X,13H PERIOD, (SEC)/(F8.4,12X,F8.4,4,F20.3))  INCN7440
118 FORMAT (/11H OCEAN WATER)  INCN7450
119 FORMAT (32H MOMENT CALC. CONTROL PARAMETERS4I5,F8.3,I5)  INCN7460
120 FORMAT (5X,7F10.0)  INCN7470
121 FORMAT (22F MOMENT CALC. AT X OF11F10.3)  INCN7480
122 FORMAT (22H MOMENT CALC. AT Y OF11F10.3)  INCN7490
123 FORMAT (33H1SES MOTIONS AND LOADS PROGRAM - 2044/)  INCN7500
124 FORMAT (23H START AND FINISH TIMES2F10.2/22H INITIAL TIME INTERVAL INCN7510
1F12.4/18H START PRINTING AT11F10.2/22H INITIAL TIME INTERVAL INCN7520
1F12.4/18H START PRINTING AT8.2,17H IN INCREMENTS OFF.8.2)  INCN7530
125 FORMAT (24H INTERMEDIATE PRINT TAGS16I5)  INCN7540
126 FORMAT (39H NO. OF STATE EQUATIONS, AND TOLERANCES15/(10X,10E12.2))INCN7550
1)  INCN7560
127 FORMAT (23H CRITICAL FROUDE NUMBERF15.4,5X,19H TOTAL CRAFT LENGTHF15.4)  INCN7570
115.4)  INCN7580
128 FORMAT (34H PLANEUM, LENGTH AND WIDTH AT WATER2F12.4)  INCN7590
129 FORMAT (34H PLANEUM, LENGTH AND WIDTH AT HULL 2F12.4/35H PLANEUM, CF INCN7600
1INTER OF PRESSURE AT HULLF12.4/23H PLANEUM, NOMINAL VOLUMEF12.1,10X,INCN7610
26HEIGHTF12.4)  INCN7620
130 FORMAT (/33H PROPULSION, X, Y, Z COORDINATES 3F12.4/)  INCN7630
131 FORMAT (/28H WEDDING, X, Y, Z COORDINATES 3F12.4/41H RUDDER, ON, MAXINCN7640
1, RATE, REVERSE, INITIAL 5F12.4/33H RUDDER, SPAN, AREA, CLB,T/INCN7650
2C 5F12.4/)  INCN7660
132 FORMAT (/39H INITIAL CONDITIONS, VELOCITY (KNOTS) =F7.2,5X,13HPITCH 1H (DEG) =F8.3,5X,12HDRAFT (IN) =F8.2)  INCN7670
SUBROUTINE INTEGRAL (TIME)

INTEGER CN

COMMON AIR, PINF, RHOINF, GAM

COMMON BMCO, IM, IMX, IMNY, IBMFIL, BTIME, IMT, XMD (10), YMD (7), IX, IY

COMMON EQNCQ, NFQS, TOL (20), JQK

COMMON KSWITCH, ITRST

COMMON MASSCST, AM, AIXX, A1XY, A1ZZ, AIXZ, AIMAX, G, WEIGHT, RHO, NMASS, AM, INT

1 (1201), X1 (1201), Y1 (1201), Z1 (1201), X5, ZS, HRHO

COMMON OPTION, ISDOF, ISRG, ITRIM, IDIA

COMMON PILENUM, XLB, XRB, ARW, BUBHGT, DVBWDT, VMDOT, PMDOT

COMMON PRIME, STIME, FTIME, DELT, DELOPT, TPRINT

COMMON PROMO1, PROMO2, PROMO3, PROMO4, PROMO5, PROMO6, PROMO7

COMMON PRINT, ON, IACCEL, IVFL, ITARJ, ISIDWL, IBOWSL, ITRANS, IWAVES, INT

RUD, IPROP, IASROD, IRS

COMMON STABLE, SI3 (4), ISTAB

COMMON STEP, STEP2

COMMON VALOLD, YOLD (20)

COMMON VARBLE, VAL (40)

EQUIVALENCE (VAL (1), X), (VAL (2), Y (1)),

DIMENSION Y (20), ERROR (20)

REAL K1 (20), K2 (20), K3 (20), K4 (20), K5 (20)

DATA IPASS / 0 /

STEP2 = 1.0

PB = VAL (24)

BMASS = Y (10)

IF ((TIME + DELOPT) .LE. TPRINT) GO TO 1

DELT = DELOPT

DELT = TPRINT - TIME

IPASS = 1

X = TIME

DO 2 J = 1, NEOS

Y (J) = YOLD (J)

2 CONTINUE
ITHRST = 1
CALL RHS (K1)
ITHRST = 2
INT = 0
IF (TACCEL .NE. ON) GO TO 3
ACCLAT = (K1(2)+Y(1)*Y(6))/G
WRITE (6,20) ACCLAT,DELT
3 ON = 2
4 H = DELT/3.
X = TIME+H
C 5 DO 5 J=1,NEOS
Y(J) = YOLD(J)+H*K1(J)
C 6 CALL RHS (K2)
C 7 DO 6 J=1,NEOS
Y(J) = YOLD(J)+.5*H*(K1(J)+K2(J))
C 8 CALL RHS (K3)
X = TIME+.5*DELT
C 9 DO 7 J=1,NEOS
Y(J) = YOLD(J)+.375*H*(K1(J)+3.*K3(J))
C 10 CALL RHS (K4)
X = TIME+DELT
C 11 DO 8 J=1,NEOS
Y(J) = YOLD(J)+.5*H*(3.*K1(J)-9.*K3(J)+12.*K4(J))
C 12 CALL RHS (K5)
IF (JCC.EQ.1) GO TO 9
C 13 DO 9 J=1,NEOS
ERROR(J) = (K1(J)-4.5*K3(J)+4.*K4(J)-.5*K5(J))*H/5.0
IF (ABS(ERROR(J)).GT.TOL(J)) GO TO 15
9 CONTINUE
C 14 DO 10 J=1,NEOS
Y(J) = YOLD(J)+.5*H*(K1(J)+4.*K4(J)+K5(J))
C 15 YOLD(J) = Y(J)
C 16 TIME = TIME+DELT
IF (IPASS.EQ.1) GO TO 14
IF (JCC.EQ.1) GO TO 12
C
INT 360
INT 370
INT 380
INT 390
INT 400
INT 410
INT 420
INT 430
INT 440
INT 450
INT 460
INT 470
INT 480
INT 490
INT 500
INT 510
INT 520
INT 530
INT 540
INT 550
INT 560
INT 570
INT 580
INT 590
INT 600
INT 610
INT 620
INT 630
INT 640
INT 650
INT 660
INT 670
INT 680
INT 690
INT 700
INT 710
INT 720
INT 730
INT 740
INT 750
INT 760
INT 770
INT 780
INT 790
INT 800
INT 810
INT 820
INT 830
DO 11 J=1,NEQS
TR (ABS(ERROR(J))).GT.TOL(J)/16.) GO TO 13
11 CONTINUE
C
DELT = 2.*DELT
IF (DELT.GT.DELPNT) DELT=DELPN
RETURN
12 STEP2 = DELT
GO TO 12
14 DELT = DF
IPASS = 0
GO TO 12
15 DELT = DELT/2.
IF (DELT.LT.1.E-6) GO TO 18
IF (JQQ,EQ,2) GO TO 17
WRITE (6,22) TIME,DELT,J,ERROR(J),TOL(J)
16 IPASS = 0
GO TO 4
17 STEP1 = DELT*2.0
IF (STEP1.LT.STEP2) STEP2=STEP1
GO TO 16
18 WRITE (6,21) TIME,DELT,(K1(J),J=1,NEQS),VAL
STOP
C
19 FORMAT (/10X,23HINTGRAL TIME,DELT,K1,VAL/2E15.4/2(6E15.4/),5(8E15.4/)
1)/)
20 FORMAT (1H0,9X,3HTOTAL LATERAL ACCELERATION (G) = F12.4,12X,5HDT
1= E15.4)
21 FORMAT (1H1,10X,44HDETA TIME LESS THAN 1.0E-6 - - JOB STOPS)
22 FORMAT (/10X,5HINT-J2E30.5,15,2E20.5)
END
C
SUBROUTINE PROP
PROP 10
PROP 20
PROP 30
PROP 40
PROP 50
PROP 60
PROP 70
PROP 80
PROP 90
PROP 100
PROP 110
PROP 120
PROP 130
EQUVALENCE (VAL(1),TIME), (VAL(2),U), (VAL(3),V), (VAL(4),W), (VAPROP 140
1L(5),P), (VAL(6),Q), (VAL(7),R), (VAL(8),PHI), (VAL(9),THETA), (VAPROP 150
INT 840
INT 850
INT 860
INT 870
INT 880
INT 890
INT 900
INT 910
INT 920
INT 930
INT 940
INT 950
INT 960
INT 970
INT 980
INT 990
INT 1000
INT 1010
INT 1020
INT 1030
INT 1040
INT 1050
INT 1060
INT 1070
INT 1080
INT 1090
INT 1100
INT 1110
INT 1120
INT 1130
INT 1140
INT 1150
C
INTEGER ON
COMMON /CCNST/ PI,RAD,U0
COMMON /FPROP/ FX,FY,FZ,FK,FM,FN
COMMON /ENGINE/ NPS,NPP,THST(25),THSTP(25),XP,YP,ZP,STHS,STHP,TIPPROP 60
PROP 70
PROP 80
PROP 90
PROP 100
PROP 110
PROP 120
PROP 130
EQUVALENCE (VAL(1),TIME), (VAL(2),U), (VAL(3),V), (VAL(4),W), (VAPROP 140
1L(5),P), (VAL(6),Q), (VAL(7),R), (VAL(8),PHI), (VAL(9),THETA), (VAPROP 150
\[2L(10), \lambda), (VAR(11), RMASS), (VAR(21), X), (VAR(22), Y), (VAR(23), PSI)\]
\[3, (VAR(24), PB)\]
DIMENSION THS(1), THP(1), TS(1), TPL(1), RUD(1), TR(1)
EQUIVALENCE (THSTS(1), THS(1)), (THSTP(1), THP(1)), (TIS(1), TS(1)),
1(TIP(1), TP(1)), (TIR(1), TR(1)), (DELRUD(1), RUD(1))

C
FX = 0.0
FY = 0.0
FZ = 0.0
FK = 0.0
FM = 0.0
FN = 0.0
TL = TIME
IF (NPR.EQ.0.0) GO TO 1
RUCANG = FG1(TL, NPR, TR, RUD, IR)
RUCANG = RUCANG/RAD

C C
CALCULATE THRUSTS AND MOMENTS INDIVIDUALLY

GO TO 2
1 RUDANG = DELRUD(1)
RUDANG = RUDANG/RAD
2 CD = CC(S(RUDANG)
SD = SIN(RUDANG)
IF (NPS.EQ.0.0) GO TO 3
THSS = FG1(TL, NPS, TS, THS, IS)
GC TO 4
3 THSS = THSTS(1)
4 IF (NPP.EQ.0.0) GO TO 5
THSP = FG1(TL, NPP, TP, THP, IP)
GC TO 6
5 THSP = THSTP(1)
6 STHSTS = STHS*THSS
STHSTP = STHP*THSP
FXS = THSS*CD-STHSTS*SD
FPX = THSTP*CD-STHSTP*SD
FS = -STHSTS*CD+STHSTS*SD
FYP = STHSTP*CD+SD*THSP
FZ = -THSS*THETA*CD+STHSTS*SD*PHI
FZP = -THSP*THETA*CD-STHSTP*SD*PHI
FX = FXP+FXS
FY = FYP+FYS
FZ = FZP+FZS
FKP = -FZP*YP-FYP*ZP
FPM = FZP*(-XP)+FYP*ZP
FMP = FZP*(-XP)+FYP*ZP
FM = FMS+FMP
FNS = -FXS*YP-FYS*(-XP)
FNP = FXP*YP-FYP*(-XP)
FN = FNS+FNP
IF (IPROP,NE.,ON) RETURN
WRITE (6,7) FX, FY, FZ, FK, FM, FN
RETURN
END

7 FORMAT (/10X,22HFRPOPF, FX, FY, FZ, FK, FM, FN/6F15.4)

SUBROUTINE RUDDER

INTEGER CN
COMMON /CONST/ PI, RAD, U0
COMMON /FRUD/ FX, FY, FZ, FK, FM, FN
COMMON /MASSES/ AM, AIXX, AIYY, AIYZ, AIXZ, AIMAX, G, WEIGHT, RHO, NMASS
1(I(201), XI(201), YI(201), ZI(201), XS, ZS, HRHO)
COMMON /PROMOD/ PROMO1, PROMO2, PROMO3, PROMO4, PROMO5, PROMO6, PROMO7
COMMON /PRINT/ ON, IACCFL, IVEL, ITRAJ, ISIDW, IBOWSL, IESTNSL, IWAVER, IRUD
IPUD, IPROP, IAEOD, IIRS
COMMON /RUDDR/ NPR, DELRUD(25), XR, YR, ZR, IRS, TL, RSPAN, RAREA, RASPR, RRUD
1CLB, RTC, RUDANG, TIR(25)
COMMON /VARBLE/ VAL(40)
EQUIVALENCE (VAL(1), TIME), (VAL(2), U), (VAL(3), V), (VAL(4), W), (VARUD
1L(5), P), (VAL(6), C), (VAL(7), R), (VAL(8), PHI), (VAL(9), THETA), (VARUD
2L(10), Z), (VAL(11), BMAS), (VAL(21), X), (VAL(22), Y), (VAL(23), PSI)
3, (VAL(24), PB)
EQUIVALENCE (DFLPUD(1), RUD(1)), (TIP(1), TR(1))
DIMENSION RUD(1), TR(1)
EQUIVALENCE (VAL(18), FAPWR)
DATA ENU/I.28E-5/
CALCULATE PROGRAMMED RUDDER DEFLECTION

t = TIME
IF (NPR.EQ.0.0) GO TO 1
GO TO 2
1 RUDANG = DELRUD(1)
RUDANG = RUDANG/RAD
GO TO 3
2 RUDANG = FGL(TL, NPR, TR, RUD, IR)
RUDANG = RUDANG/RAD
SIDE FCRCE ON RUDDER
3 DSR = Z+ZS-XP*THETA
ENDFAC = (1.+DSR/(DSR+RSPAN))
VH = V + XR ** ZR ** P
QQ = HRRD ** U * U * RAREA
EFFANG = RUDANG - ENDFAC * VH / U
FY = 2 * QQ ** ENDFAC * RCLB ** EFFANG

** DRAG FORCE ON PUDDER **
REY = U ** (RAREA / RSPAN) / ENU
CFR = 427 / (ALOG10(REY - 407) ** 2.64
PB = PI / 8.
CD = 2 * CFR ** PB * RCT * RCT * (1 + G * RSPAN / (U * U)) + RCLB ** EFFANG ** EFFANG
FX = -2 * CD ** RARE + HRRD ** U * U
FZ = 0.
FK = - ZR * FY
FM = FX * ZR
FN = XR * FY
IF (IRUD, NE, ON) RETURN
WRITE (6, 4) FX, FY, FZ, FK, FM, FN

RETURN

4 FORMAT (10X, 24H#PUDDER FX, FY, FZ, FK, FM, FN/6E15.4)

END

SUBROUTINE RHS (VALUF)
INTEGER CN
COMMON /AIR/ PINF, RHINO, GAM
COMMON /ATPIX/ AMASSI, AYYI, DIXX, DIXZ, DIZZ
COMMON /BMC0/ IMM, IMNX, IMNY, IBMFIL, BTIME, IMT, XMI(10), YMI(7), IX, IY
COMMON /COLUMN/ IVERT, ILATRL
COMMON /CONST/ PI, RAD, UO
COMMON /CNTRL/ CONTW, CONTQ, CONTTH, QMULT, LOUVER, ACONTZ, ACONTW, ZEQUIHR
COMMON /ENGINE/ NPS, NPP, THST(25), THSTP(25), XP, YP, ZP, STHS, STHP, TIP
COMMON /FANMAP/ QIN, QBFAN(25), QMFAN(25), QSFA(25), ENBFAN, ENMFAN, ENRF
COMMON /FRM/ BPM, MRP, PRTM, NPTSM, NPTSS, PBFAN(25), PMFAN(25), PSFAN(25)
COMMON /FTHR/ DELR(25), NB, TMES(25), DETA(25), NS
COMMON /FAERO/ FZ, FZAED, FTAED, FEAED, FMAED, FNAED
COMMON /FORS/ FZBS, FYBS, FZBS, FKBS, FMBS, FNBS, QLBS
COMMON /FORMS/ FXSS, FYSS, FZSS, FKSS, FMSS, FNSS, QLSS, FMS
COMMON /PROP/ FXP, FZP, FKP, FNP
COMMON /RUDE/ FM, FMCRT
COMMON /FU/ FXU, FRYU, FZRU, FKPU, FMRU, FNRU
COMMON /GROH/ XBO, XCP, ZCP

RHS 10
RHS 20
RHS 30
RHS 40
RHS 50
RHS 60
RHS 70
RHS 80
RHS 90
100
RHS 110
RHS 120
RHS 130
RHS 140
RHS 150
RHS 160
RHS 170
RHS 180
RHS 200
RHS 210
RHS 220
RHS 230
**COMMON** /GEOBS/ DETABX(11), DETABT(11), ARM1B(10), ARM2B(10), DFBS(10)RHS 240
1, TSKE(10) RHS 250
**COMMON** /GEOSS/ DETADX(11), DETADT(11), ARM1S(10), DFSS(10), TSKIS(10)RHS 260
1, ARM2S(10) RHS 270
**COMMON** /KSWITCH/ 1TRST RHS 280
**COMMON** /MASS/ AM, AIXX, AIXY, AIYZ, AMAX, G, WEIGHT, RH0, NMASS, AMRHS 290
1, YI(201), ZI(201), XX, ZZ, HRHOD RHS 300
**COMMON** /MSIDW/ DF(2, 10), DSWAV(2, 10), FXH(2), FYH(2), FZH(2), FMH(2), FNHRHS 310
1H(2), VFY(2), VFZ(2), FXVRHS 320
**COMMON** /MWA/ FWX(2), FWY(2), FZW(2), FKW(2), FMW(2), FNW(2)RHS 330
**COMMON** /OPTION/ I3DDF, ISRGE, ITIN, IDIA RHS 340
**COMMON** /PLENUM/ XLBW, XBBW, ABW, BUBHG, DVWDOT, VMDOT, CMMRHS 350
**COMMON** /PRIME/ STIME, FTIME, DELT, DELLNT, TPRINT RHS 360
**COMMON** /PRINT/ ON, IACCEL, IVEL, ITRA, ISIDW, IBOWSL, ISTNL, IWAVERHS 370
1RUD, IREP, IAEED, IPHS RHS 380
**COMMON** /PRMOD/ PROMO1, PROMO2, PROMO3, PROMO4, PROMO5, PROMO6, PROMO7RHS 390
**COMMON** /PWA/ FNCON, PVCON RHS 400
**COMMON** /RUDOR/ NPR, DELFUD(25), XR, YR, ZR, IRS, TL, RSPAN, RAREA, RASPRHRHS 410
1CLB, RTC, RUCANG, TIR(25) RHS 420
**COMMON** /SIDE/ FXSW, FYSW, FZSW, FKSW, FSMW, FNFW, ALSW, YSW, XLSW, CFSW, CDSRHS 430
1W, VAREA, VCHRD, VSPAN, VANGLE, VCGS, VX, YV, ZV, AVBMWS, DELX, VTCRHS 440
**COMMON** /SLOPE/ WATSLP, XPW, XLXPW, PWVHT, XPWVXRHS 450
**COMMON** /SOFTBS/ XBF, PB5, SINBS, COSBS, XBS, ZBS, DELYBS, DPBS, ELMAXB, YAVRHS 460
1GIB(10), CENCAB RHS 470
**COMMON** /SOFTSS/ XLF, PSS, SINTH, COSTH, XSS, ZSS, DELYSS, DPSS, ELMAXSS, YAVRHS 480
1GSS(10) RHS 490
**COMMON** /VALOLD/ YOLD(20) RHS 500
**COMMON** /VARBLE/ VAL(40) RHS 510
**COMMON** /WAVE/ ETA(4, 11), AW(20), OMEGA(20), DVOLW, NWAVER, BETA, FXWAV, FYHRHS 520
1WAVE, FZW, FWK, FMWAV, FNWAV, ZBAR, PHIBAR, THEBAR, COSB, SINF, SINB, PRBSRHS 530
2BAR RHS 540
**EQUIVALENCE** (VAL(1), TIME), (VAL(2), U), (VAL(3), V), (VAL(4), W), (VARHS 550
1L(5), P), (VAL(6), Q), (VAL(7), R), (VAL(8), PHI), (VAL(9), THETA), (VARHS 560
2L(10), Z), (VAL(11), AMASS), (VAL(21), X), (VAL(22), Y), (VAL(23), PSI)RHS 570
3, (VAL(24), PP) RHS 580
**EQUIVALENCE** (VAL(18), FNPWR) RHS 590
**DIMENSION** ACCEL(3), ANGACL(3) RHS 600
**DIMENSION** GF(6), VALUE(20) RHS 610
**DATA** CCF/4.9E-07/

**DO 1 J=1,20**
1 **VALUE(J) = 0.0** **RHS 620**

**CALCULATION OF BUBBLE WAVE MAKING DRAG** RHS 630

**AB = ABW-(ABW-(XL*WIDTH))*(ZS+Z)/BUBHGT** RHS 640
FN = U/FNCON
CF = .37/(FN**1.5655981)
FPWAV = -FWVCON*PBAR*CF
WATSLP = -FXPS/WGHT
VOL = VAL(12)
PB = VAL(13)
PBAR = VAL(13) - P_INF
PSS = FB+DPES
ABPB = PBAR*AB
CALCULATION OF BUBBLE WAVE MAKING DPAG
CLS = CSWS*ALSW*FLOW
CALL BOWSL
CALL STNSL
SUMX = FXBS+FXSS+FSXW+FXRUD+FXP+FXWAV+FAED+FXPWAV
IF (ITHST .NE. ITRIM) GO TO 2
THSTP1 = THSTP(1) - SUMX*.5
THSTP(1) = THSTP(1) - SUMX*.5
THST = THSTP(1) + THSTP(1)
SUMX = 0.0
2 CONTINUE
SUMY = FYBS+FYSS+FSWY+FRUD+FYP+FZWP+FZWA+FZED
SUMZ = WGT-ABBP+BFB+BFXS+FSXW+FXRUD+FZP+FXWAV+FZED
SUMK = FKBS+FKSS+FKSW+FKRUD+FKP+FKWAV+FKAED+ABPB*PHI*(-Z)
CALCULATION OF EFFECTIVE CENTER OF PRESSURE
UU = U*0.5921-30.0
XCPU = XCP+0.001975*UU*UU-0.974
XCP = SHXYAX(XCPU,ZCP,THETA,PI)
FMUB = ABBP*XCP
SUMM = FMBS+FMSS+FSG+FRUD+FMP+FMWAV+FAED+FMUB-FXPWAV*ZS
FWAVZ = FXPWAV*ZS
SUMN = FNBS+FNSS+FNSW+FRUD+FNP+FWAV+FNAED
IF (I3DOF .NE. 1) GO TO 3
SUMM = 0.0
3 CONTINUE
SUMM = 0.0
VALUE(1) = SUMX*AMASSI
VALUE(2) = SUMY*AMASSI-R*U
VALUE(3) = SUMZ*AMASSI
VALUE(4) = SUMK*DIZZ+SUMN*DIZZ
VALUE(5) = SUMMAIYI
VALUE(6) = SUMN*DIXB+SUMK*DIXZ
VALUE(7) = P
VALUE(8) = Q
VALUE(9) = W
IF (I3COF.EQ.1) GO TO 6

BUBBLE PRESSURE EQUATION

COUT = QLBS*QLSS*QLSW

CALL FAN
VALUE(10) = RHOINF*(QIN-QOUT)
VALUE(11) = -((X1*W1D1)+ABW)*.5*V-DVWDOT
VALUE(12) = VAL(13)*GAM*((VALUE(10)/VAL(11))-(VALUE(11)/VAL(12)))
IF (IDIA.EQ.1) GO TO 4
GO TO 5
4 CONTINUE

MEMBRANE STUDY

VAL(12)=VAL(12)+0.5175*VAL(13)
VALUE(12)=VAL(13)*GAM*((VALUE(10)/VAL(11))-(VALUE(11)/VAL(12)))
5 CONTINUE
GO TO 7
6 CONTINUE
VALUE(10) = 0.0
7 CONTINUE

WRITE DATA FILE FOR MOMENT AND SHEAR CALCULATIONS IF REQUIRED

IF (IMNTNE1) GO TO 8
NBS = NSTA(3)-1
NSS = NSTA(4)-1
NSL = NSS/2+1
WRITE (IMFIL) (VAL(I),I=1,24),ZBAR,PHIBAR,THEBAR,FXW,FWY,FWZ,FKW,RHS = 1510
1FMW,FMW,(VALUE(I),I=1,10),DF,DSWV,FXH,FYH,FZHK,FMY,FMH,FMH,FDV,FDVRS
2,FXRUD,FXRUD,FPF,FPF,FPF,FZSS,FMSS,FXBS,FXBS,FXBS,FXAED,FXAED,FXAED
3D,FXAED,FXAED,FXPFW,FXSS,FKBS,FMBS,FMBS,FMBS,FMBS,FMBS,(TSKIS(I),DFSS(I),I=1,RHS = 1570
4NSL+NSS),(TSKIB(I),DFBS(I),ARMIB(I),ARMIB(I),I=1,NBS)
8 CONTINUE

CONSTANT LONGITUDINAL VELOCITY (U)

IF (ISRGE.EQ.1) VALUE(I)=0.0
IF (ON.NE.1) RETURN

DO 9 I=1,3
ACCEL(I) = VALUE(I)/G
ANGACL(I) = VALUE(I+3)*RAD
9 CONTINUE
19 FORMAT (/10X,24H TOTAL FORCES AND MOMENTS 6E12.4/10X,24H ACCELERATION RHS 2170
1S G,DEG/SEC26E12.4)
RHS 2180
20 FORMAT (/10X,16HBOW ACCEL. (G) =F12.4,21H STERN ACCEL. (G) =F12. RHS 2190
14)
RHS 2200
END

SUBROUTINE SAM

C
WRITE (6,1)
RETURN
C
1 FORMAT (IH1, 'YOU HAVE CALLED A DUMMY SAM SUBROUTINE.'/
110X, 'CHANGE TO BH1SES TO USE THE SAM SUBROUTINE.')
SAM 60
END

SUBROUTINE SIDWL

C
INTEGER ON
COMMON /AIR/, PINF, RHOINE, GAM
COMMON /BMCO/, IMX, IMNY, IBMFIL, BTIME, IMT, XMIN(10), YM(7), IX, IY
COMMON /CONST/, PI, RAD, UD
COMMON /GEOM/, WIDTH, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4, 1)
101), XCP, ZCP
COMMON /GEOMSW/, XAVG(10), DS
COMMON /SWITCH/, ITPST
COMMON /MASS/, AM, AIXX, AIXY, AIZZ, AIXZ, AIMAX, G, WEIGHT, RHO, NMASS, AM
COMMON /I201/, XI201), Y201), ZI201), XS, ZS, HRH
COMMON /MIS120/, DF(2, 10), DSWAV(2, 10), FXH(2), FYH(2), FZH(2), FMH(2)
COMMON /LH2/, VFY(2), VFZ(2), FXV
COMMON /PREM/, XLB1, XBBW, AW, BURBGT, DVWDOT
COMMON /PRICE/, STIME, FTIME, DELT, DELPNT, PPRINT
COMMON /PRINT/, ON, ACCEL, IVEL, ITRAJ, ISDWL, IBOWS, IBSNL, IWAVS, IS
COMMON /RUD/, IPRF, IAERO, IRH
COMMON /PROMOD/, PROMO1, PROMO2, PROMO3, PROMO4, PROMO5, PROMO6, PROMO7
COMMON /SIDF/, FX, FY, FZ, FM, FN, ALSW, YSW, XLSW, CFSSW, CDSW, VAREA, VCROSS
COMMON /STP/, VS, VANG, VCO, VX, VY, VZ, AVB, ML, DELX, VTC
COMMON /SLOPE/, WATSL, WPXW, XLPW, PWHT, WPXVX
COMMON /VARBLE/, VAL(40)
COMMON /WAFF/, ETA(4, 11), AW(20), OMEGA(20), DVLW, NWAVE, BETAF, FXWAV, FYSDW
COMMON /WAVF/, FWAV, FWAV, WMVW, FMWV, FWAV, ZBAP, PHIBAR, THEBAR, TC, COSBET, SINFET, PBSW
COMMON /WAT/ AB, /NACL, CAL, SAL, NDS, DDS, DDS, NTH, DTH, STH, NBR, DBB, SRB, ACSDW
11(20, 5, 7), AC2(20, 5, 7), AC3(20, 5, 7), AC4(20, 5, 7), AC5(20, 5, 7), AC6(20, 5, 7)
2(3), AC7(20, 5, 7), AC0(20, 5, 7), AC10(20, 5, 7), AC20(20, 5, 7), AC40(20, 5, 7), AC60(20, 5, 7)
3S(20, 5, 7), AS3(20, 5, 7), AS4(20, 5, 7), AS5(20, 5, 7), AS6(20, 5, 7), AS7(20, 5, 7)
4(20, 5, 7), ASC(20, 5, 7), ASO0(20, 5, 7), ASB(20, 5, 7), BB(36), XREF, RX
COMMON /VARBLE/, VAL(40)
COMMON /WAVTAB/, NAL, CAL, SAL, NDS, DDS, DDS, NTH, DTH, STH, NBR, DBB, SRB, ACSDW
11(20, 5, 7), AC2(20, 5, 7), AC3(20, 5, 7), AC4(20, 5, 7), AC5(20, 5, 7), AC6(20, 5, 7)
2(3), AC7(20, 5, 7), AC0(20, 5, 7), AC10(20, 5, 7), AC20(20, 5, 7), AC40(20, 5, 7), AC60(20, 5, 7)
3S(20, 5, 7), AS3(20, 5, 7), AS4(20, 5, 7), AS5(20, 5, 7), AS6(20, 5, 7), AS7(20, 5, 7)
4(20, 5, 7), ASC(20, 5, 7), ASO0(20, 5, 7), ASB(20, 5, 7), BB(36), XREF, RX
COMMON /VARBLE/, VAL(40)
1, Q, (VAL(7), R), (VAL(8), PHI), (VAL(9), THETA), (VAL(10), Z), (VAL(11), Z)
1, Q, (VAL(7), R), (VAL(8), PHI), (VAL(9), THETA), (VAL(10), Z), (VAL(11), Z)
DIMENSION GAP(2,11), DSW(2,11)
DIMENSION FZHOLD(2), FZHDRP(2)
DATA ENU/1.28E-5/

C
PBAR = PB-PINF
PBHEAD = PBAR/(RHO*G)

C
GAF OR WETTED DRAFT CALCULATION

DO 5 J=1,2
N = NSTA(J)

C
DO 5 K=1,N
DD = ZS+2+YY(J,K)*PHI-XX(1,K)*THETA*ETA(J,K)
DDIN = DD-WATSLP*(XPWVS-XX(J,K))
IF (DDIN.LT.BUBHGT) GO TO 1
IF (VAL(1).TOL.DLT.DELPNT) GO TO 1
TOLD = VAL(1)
WRITE (6,14) XX(J,K),VAL(1),DD
1 CONTINUE
DSW(J,K) = (SIGN(1.,DD)+1.)*DD/2.
IF (DDIN) 2,4,4
2 IF (DSW(J,K)-PBHEAD) 3,4,4
3 GAP(J,K) = -DDIN*(1.-DSW(J,K)/PBHEAD)
GO TO 5
4 GAP(J,K) = 0.0
5 CONTINUE

C
LEAKAGE AREA

C
ALSW = 0.0

C
DO 6 J=1,2
N = NSTA(J)-1

C
DO 6 I=1,N
ALSW = ALSW+(GAP(J,I)+GAP(J,I+1))*DELX/2.
6 CONTINUE

C
CROSS-FLOW DRAG ON SIDEWALLS

C
FYD = 0.0
FKD = 0.0
FND = 0.0
C DO 7 I=1,2
N = NSTA(I)-1
C DO 7 J=1,N
DSWAV(I,J) = (DSW(I,J)+DSW(I,J+1))/2.
VREL = V+XAVG(J)*P-(2*DSWAV(I,J))/2.*P
DF(I,J) = -HRHO*CSW*VREL*ABS(VREL)*DELX*DSWAV(I,J)
FYD = FYD+DF(I,J)
FND = FND+DF(I,J)*XAVG(J)
7 FKD = FKD-(2*DSWAV(I,J))/2.*DF(I,J)
C C SET UP STERN LIMIT OF FORCE DETERMINATION
C XSS = -XS
GO TO 8
ENTRY CIDWLM
XSS = XMI(IK)
8 IP = 1.+((THETA+PAD-STH)/DTH
IP = MAXO(MINO(IP,NTH),1)
IP1 = MINO(IP1,NTH)
DTHETA = (IP-1)*DTH+STH
CIP = (THETA-RAD-DTHETA)/DTH
C C CALC REYNOLDS NO. AND DRAG COEFF.
C REY = U*XLSW/ENU
CDT = .427/(ALOG10(REY)-.407)**2.64
C C SIDEWALL FORCES, P/S
C DO 11 J=1,2
WARFA = 0.0
N = NSTA(J)-1
NI = (XSS+XS)*N/XLSW+1.5
C DO 9 I=NI,N
ZOR1 = 1.0
IF (DSWAV(J,I),EQ,0.0) ZOR1=0.0
9 WARFA = WARFA+DELX*(2.*DSWAV(J,I)+ZOR1*AVBMSW)
C FXH(J) = -HRHO*CDT*WARFA*U*U
PM1 = 2.*J-3
YLSW = PM1*YSW
DS = Z+ZS+YLSW*PHI
DSS = DS-XSS*THETA
C
FK = FK-PROMO2*YSW*YBC2*P/P1
FZH(1) = FZH(1)-PROMO2/2.*YSW*YBC2*P/P1
FZH(2) = FZH(2)-PROMO2/2.*YSW*YBC2*P/P1
IF (PROMO3.FQ.1.0) WRITE (6,15) VAL(1), FZHOLD(1), FZHOLD(2), FZHCRP(11), FZHCRP(2), FZH(1), FZH(2), FKOLD, FK
IF (TSIDWL.NE.ON) RETURN
C
DO 13 I=1,2
C
DO 13 J=1,11
GAP(I,J) = 12.0*GAP(I,J)
DSW(I,J) = 12.0*DSW(I,J)
C
WRITE (6,16) ((GAP(I,J), J=1,11), I=1,2), ((DSW(I,J), J=1,11), I=1,2), F1, F2, F3, F4, F5, F6, F7, F8, F9, F10
RETURN
C
14 FORMAT (/10X,43HCONTACT WITH TOP OF BUBBLE CHAMBER AT F7.2,14 SDWL 2450 2 FT., TIME=F7.2,19H SEC., IMMERSION=F7.2,4H FT.)
15 FORMAT (/2X,'TIME=1.1X,E15.4,2X,'OLD VERTICAL FORCES',2(5X,E15.4)/ SDWL 2470 1 25X,'NEW VERTICAL FORCES',2(5X,E15.4)/25X,'NEW VERTICAL FORCES',2(5X,E15.4)/ SDWL 2480 2ES',2(5X,E15.4)/25X,'OLD AND NEW ROLL MOMENTS',2(5X,E15.4)/ SDWL 2490 16 FORMAT (/10X,8HSIDEBALL/25H GAP (FT.) (STERN TO BOW)/14H PORT SIDE SDWL 2500 1 WALL/11F10.5/14H STBD SIDEWALL/11F10.5/37H IMMERSION DEPTH (FT.) (SDWL 2510 2 STERN TO BOW)/14H PORT SIDEWALL/11F10.5/14H STBD SIDEWALL/11F10.5/ SDWL 2520 3 10X,26+SIDEWALL FX,FY,FZ,FK,FM,FN/6F15.4) END
FUNCTION SHXYAX (X,Z,ANGYAX,PI) SHX 10
H = SQRT(X*X+Z*Z)
IF (X.EQ.0.0) GO TO 1
ARG = Z/X
ANGOLD = ATAN(ARG)
IF (ANGOLD.GE.0.0) GO TO 2
ANGNEW = ANGOLD+PI-ANGYAX
GO TO 3
1 ANGNEW = PI*.50-ANGYAX
GO TO 3
2 ANGNEW = ANGOLD-ANGYAX
3 SHXYAX = H*COS(ANGNEW)
RETURN
END
SUBROUTINE STNSL SSL 10
INTEGER ON
COMMON /AIR/ PINF,RHCINF,GAM
COMMON /CONST/ PI,RAD,UD
COMMON /FORSS/ FX, FY, FZ, FK, FM, FN, QL, FMS
COMMON /GEOM/ WIDTH, XL, XX(4, 11), YY(4, 11), NSTA(4), AR, VOLNOM, DELS(4), SSL
110, XCF, ZCP
COMMON /GEOMS/ DETADX(11), DETADT(11), ARMIS(10), DFSS(10), TSKIS(10) SSL
1, ARM25(10)
COMMON /KSWITCH/ ITHST
COMMON /LEAKER/ ALEAK, CFSS
COMMON /MASSES/ AM, AIXX, AIYY, AIZZ, AIXZ, AIMAX, G, WEIGHT, RHO, NMASS, AMSS
111(201), XI(201), YI(201), ZI(201), XS, ZS, HRHO
COMMON /PRINT/ ON, IACCEL, IVEL, ITRAJ, ISIDWL, IBOWSL, IISTNL, IWAVES, SSL
130, IUP, IPROP, IAFEROD, IRHS
COMMON /PROP/ PROM0, PROM1, PROM2, PROMP, PROMQ, PROMO, PROM6, PROM7
COMMON /SLICE/ WATSPL, XPIV, XLWXY, PILVHT, XPIVX
COMMON /SOFTEE/ XLF, PSS, SINWH, COTH, XSS, ZSS, DELYSS, DPSS, ELMAXS, YAVSSL
150, 1GS(10)
COMMON /STSLR/ CPHI, CPHID
COMMON /VAPBLE/ VAL(20)
COMMON /WAVE/ ETA(4, 11), AW(20), OMEGA(20), DVOLW, NAVE, BETAX, FNXWAV, FYSS
170, 1WAV, FZ, FWAV, FK, FMWAV, FNWAV, ZBAR, PHI, PAR, THEBAF, TC, COS, SIN, PBSSL
2BAR
EQUIVALENCE (VAL(1), TIME), (VAL(2), U), (VAL(3), V), (VAL(4), W), (VASSL
270, 1L(5), P), (VAL(6), Q), (VAL(7), R), (VAL(8), PHI), (VAL(9), THTA), (VASSL
280, 2L(10), Z), (VAL(11), AMASS), (VAL(21), X), (VAL(22), Y), (VAL(23), PSI)
3, (VAL(24), PB)
DIMENSION GAP(11), ELSSI(11), ELSKL(11), AIRL(11), CTNLS(6, 24) SSL
310, DATA ENU, GPS, HINGH1/1, 28E-5, .4644, 1.875/-
320, DATA CTNLS/0.0, 3.87, 0.9, 9.12.3, 9.15.0, 0, 0.41, 7.3, 10.5, 13.2, 15.7, 0.0SSL
330, 10.4, 3.7, 7.11.0, 13.7, 15.7, 0, 0, 0, 4.7, 8.2, 11.5, 14.5, 17.4, 0, 0, 0.5, 8.9, 9.12SSL
340, 2.4, 15.4, 18.3, 0, 0, 5.9, 8.4, 13.1, 16.1, 19.2, 0, 0, 5.8, 9.9, 13.8, 17.1, 20.6SSL
350, 21.0, 0.0, 1.1, 1.0, 6.14.7, 18.4, 21.9, 0.0, 6.4, 11.0, 15.3, 19.3, 23.3, 0.0, 6.6, 9.1SSL
360, 41.7, 10.4, 20.5, 24.5, 0.0, 6.9, 12.3, 17.6, 22.1, 25.7, 0.0, 7.2, 13.5, 19.6, 22.9SSL
370, 54.1, 20.3, 30.0, 0.0, 8.2, 15.3, 22.1, 26.3, 30.9, 0, 0, 9.6, 17.5, 24.2, 29.7, 34.6SSL
380, 60.0, 11.9, 20.0, 25.4, 25.3, 38.0, 0, 0, 14.2, 23.3, 30.0, 0.35, 8.4, 0.0, 0.0, 17.3SSL
390, 725.5, 32.5, 39.7, 53.4, 0.0, 21.6, 30.0, 0.38, 7.4, 20, 0.0, 24.6, 33.7, 41.3SSL
400, 30.45, 0.48, 7.0, 0.0, 28.3, 38.5, 45.0, 0.48, 7.4, 10.0, 35.0, 45, 0.48, 7.5, 8.7, 4SSL
410, 1, 98.7, 0.0, 40.0, 48.7, 48.7, 48.7, 48.7, 0.0, 48.7, 48.7, 48.7, 48.7, 48.7, 48.7, 0.0,$48.7, 48.7, 48.7, 48.7, 48.7, 48.7, 48.7, 48.7, 0.0SSL
420, IF (ELMKSX.CT.HINGH1) ELMAXS=HINGH1
430, DO 1 J=1,11
440, GAP(J) = 0.0
450, ELSKI(J) = 0.0
460, C
470, C
ELSkl(J) = 0.0
AIRLEN(J) = 0.0

CONTINUE

'EFDEP' IS THE EFFECTIVE LIFTING DEPTH OF THE STERN SEAL

EFDEP = 6.0
MM = EFDEP
ALSS = 0.0
FX = 0.0
FZ = 0.0
FK = 0.0
FM = 0.0
FN = 0.0
AGAP1 = 0.0
AGAP2 = 0.0
AGAP1 = 0.0
DELP = PSS-PB
IF (DELP,LT,0.0) DELP=0.0
PBAR = PB-PINF

CALCULATE ELSKI HERE.

SINDIF = SINTH*COSTH*THETA
COSDIF = COSTH*SINTH*THETA
XI = XSS+ZSS*THETA-XLF*SINDIF
ZI = (-Z-ZSS+XSS*THETA-ELMAXS*COS(THETA))

CALCULATE GAP HERE.

N = NSTA(4)

DC 2,K=1,N
ELSkl(K) = (ETA(4,K)-DETADX(K)*(XX(4,K)-XI)-ZI)+YY(4,K)*PHI-XPWV*W
1LITSLP
IF (ELSkl(K),GT,HIGHT) ELSkl(K)=HIGHT
ELSkl(K) = ELSkl(K)+GPS
IF (ELSKl(K),LE,HIGHT) ELSkl(K)=HIGHT
IF (ELSKl(K),LT,(HIGHT-ELMAXS)) ELSkl(K)=HIGHT-ELMAXS
GAP(K) = -ELSKl(K)+(HIGHT-ELMAXS)
IF (GAP(K),LT,0.0) GAP(K)=0.0
MM1 = ELSkl(K)*12.0
MM2 = MM1+1
MM3 = MM2+1
DLINC = ELSkl(K)*12.0-MM1
STNSL1 = CTNSL(MM,MM2)
STNSL2 = CTNSL(MM,MM3)
AIRELEN(K) = (((STNSL2-STNSL1)*DLINC+STNSL1)/12.0
2 CONTINUE
C
N = NSTA(4)-1
C
DO 5 J=1,N
ELSITA = (ELSIT(J+1)+ELSIT(J))*0.5
ELSITL = (ELSITL(J+1)+ELSITL(J))*0.5
AIRELV = (AIRELEN(J+1)+AIRELEN(J))*0.5
AGAP = ELSITL-ELSITA
AGAP1 = AGAP
IF (AGAP.LT.GPS) AGAP=GPS
IF (AGAP.LT.GPS) AGAP1=GPS
ARM1S(J) = XX(4,J)+ELSITL*0.5
ARM2S(J) = ZS-ELSITL
DFSSL(J) = -DELP*DELYS*AILAV/(GPS/AGAP)**2
GO TO 6
IF (AILAV.GE.0.0) GO TO 3
ARG = 5*RH0+U*AILAV*DELYS
RESK1 = U*AILAV/ENJ
CDTSKI = 427/((ALOG10(RESK1)-.407)**2.64
TSKIS(J) = -ARG*CDTSKI
GO TO 4
3 TSKIS(J) = 0.0
4 CONTINUE
5 CONTINUE
THE FOLLOWING CARD REMOVES WATER DRAG EFFECTS OF STERN SEAL
C
TSKIS(J) = 0.0
GO TO 4
3 TSKIS(J) = 0.0
4 CONTINUE
5 CONTINUE
FX = FX+TSKIS(J)
FZ = FZ+DFSSL(J)
FK = FK+DFSSL(J)*YAVGS(J)
FM = FM+DFSSL(J)*ARM1S(J)+TSKIS(J)*ARM2S(J)
FN = FN+TSKIS(J)*YAVGS(J)
ALSS = ALSS+(GAP(J)+GAP(J+1))*DELYS*0.5
AGAP2 = AGAP2+AGAP1
AGAPA1 = AGAP2/J
5 CONTINUE
C
ALSS = ALSS*ALEAK*(AGAPA1/GPS)
SQFAC = SQRT(2.*ABS(PBAR)/RH01NF)
QL = CFSSL*ALSS*SQFAC*SIGN(1.*PBAR)
IF (ISTNSL.NE.ON) RETURN
WRITE (*,6) GAP, AIRELEN, FX, FY, FZ, FK, FM, FN
C
RETURN
C
6 FORMAT (/12H STERN SEAL/26H GAP (FT.) PORT TO STBD. /11F11.3/28SS
C
0
FUNCTION T1 (X)
1 T1 = X*(1.-X*X/10.0)/3.
RETURN
2 T1 = (SIN(X)-X*COS(X))/(X*X)
RETURN
END

FUNCTION T2 (X)
1 T2 = 1.-X*X/6.
RETURN
2 T2 = SIN(X)/X
RETURN
END

INTEGER CN
COMMON /BCMD/) IMM, IMNX, IMDY, IBMIL, BTIME, IMT, XMI(10), YMI(7), IX, IY WAVS 10
COMMON /HCCM/) PI, RAD, 00 WAVS 20
COMMON /GOEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 30
WAVS 40
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 50
WAVS 60
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 70
WAVS 80
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 90
WAVS 100
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 110
WAVS 120
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 130
WAVS 140
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 150
WAVS 160
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 170
WAVS 180
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 190
WAVS 200
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 210
WAVS 220
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 230
WAVS 240
COMMON /GEM/) WIDX, XL, XX(4, 11), YY(4, 11), NSTA(4), AB, VOLNOM, DELS(4), WAVS 250
WAVS 260
FMW(J) = 0.0
FNW(J) = 0.0
4 CONTINUE
C XJS = -XS
   IF (IMT.EQ.2) XJS = XMI(IX)
   IP = 1 + (THEBAR*RAD-STH)/DTH
   IP = MAXO(MINO(IP, NTH), 1)
   IIP = MNO(IP+1, NTH)
   DTHETA = (IP-1)*DTH+STH
   DIP = (THEBAR*RAD-DTHETA)/DTH
   TIME RISE FACTOR FOR WAVE AMPLITUDE
   AMPFAC = 1. - EXP(-TIME/AMPTC)
   DO 11 I = 1, NWAVE
      GMI = CMEGA(I)
      GM2 = CM1*CM1
      XWK = CM2/G
      AA = AWI(I)*AMPFAC
      FT = GMI*TIME*XWK*FO
      AL = XWK*COCAM
      IAA = 1 + (ABS(AL)-SAL)/DAL
      IAA = MAXO(MINO(IAA, NAL), 1)
      IAA1 = MNO(IAA+1, NAL)
      DAA = (IAA-1)*DAL+DAL
      DIA = (ABS(AL)-DAA)/DAL
      SALP = SIGN(1., AL)
C WAVE FORCES AND MOMENTS ON THE SIDEWALLS
C
   DO 7 J = 1, 2
      YLSW = (2*J-3)*YSW
      WE = FT*XWK*SIGAM*YLSW
      ST = SIN(WE)
      CT = COS(WE)
      DS = ZBAR+ZS*YLSW*PHIBAR
      DSR = DS-(XREF-XS)*THEBAR
      ID = 1 + (DS*12-SDS)/DDS
      ID = MAXO(MINO(ID, NDS), 1)
      DDDR = (ID-1)*DDS+DDS
      CID = (DSR*12-DDDR)/DDS
      ID1 = MNO(ID+1, NDS)
      DSS = DS-XSS*THEBAR
      ZOR1 = (SIGN(1., DSS)+1.)*0.5
      DSS = DSS*ZOR1
      DSS = 1.5*(DSS-SBB)/DBB
      IDSS = MNO(NBB, IDSS)
BS = BB(IDSS)
CR = CCSXMK*COSAM*XSS)
A33S = (RH0*PI*BS*BS)*0.125
SK = SINXMK*COSAM*XSS)
A22S = (RH0*.4*PI*DS*DS)*0.5
A42S = 0.0

INTERPRETATION OF WAVE TABLES

K = 1
L = [AA
CONTINUE
BC0 = AC0(L, ID, IP)
BC00 = AC00(L, ID, IP)
BC1 = AC1(L, ID, IP)
BC2 = AC2(L, ID, IP)
BC3 = AC3(L, ID, IP)
BC4 = AC4(L, ID, IP)
BC5 = AC5(L, ID, IP)
BC6 = AC6(L, ID, IP)
BC7 = AC7(L, ID, IP)
BC8 = AC8(L, ID, IP)
BS0 = AS0(L, ID, IP)
BS1 = AS1(L, ID, IP)
BS2 = AS2(L, ID, IP)
BS3 = AS3(L, ID, IP)
BS4 = AS4(L, ID, IP)
BS5 = AS5(L, ID, IP)
BS6 = AS6(L, ID, IP)
BS7 = AS7(L, ID, IP)
BS8 = AS8(L, ID, IP)

WC0(K) = BC0 + ID*(AC0(L, ID1, IP) - BC0) + DIP*(AC0(L, ID, IP1) - BC0 + ID)*(A waters 1550
1C0(L, ID1, IP1) - AC0(L, ID, IP1) - AC0(L, ID1, IP) + BC0)
WC00(K) = BC00 + ID*(AC00(L, ID1, IP) - BC00) + DIP*(AC00(L, ID, IP1) - BC00 + ID)*(A waters 1550
1DIP*(AC00(L, ID1, IP1) - AC00(L, ID, IP1) - AC00(L, ID1, IP) + BC00)
WC1(K) = BC1 + ID*(AC1(L, ID1, IP) - BC1) + DIP*(AC1(L, ID, IP1) - BC1 + ID)*(A waters 1590
1C1(L, ID1, IP1) - AC1(L, ID, IP1) - AC1(L, ID1, IP) + BC1)
WC2(K) = BC2 + ID*(AC2(L, ID1, IP) - BC2) + DIP*(AC2(L, ID, IP1) - BC2 + ID)*(A waters 1600
1C2(L, ID1, IP1) - AC2(L, ID, IP1) - AC2(L, ID1, IP) + BC2)
WC3(K) = BC3 + ID*(AC3(L, ID1, IP) - BC3) + DIP*(AC3(L, ID, IP1) - BC3 + ID)*(A waters 1630
1C3(L, ID1, IP1) - AC3(L, ID, IP1) - AC3(L, ID1, IP) + BC3)
WC4(K) = BC4 + ID*(AC4(L, ID1, IP) - BC4) + DIP*(AC4(L, ID, IP1) - BC4 + ID)*(A waters 1650
1C4(L, ID1, IP1) - AC4(L, ID, IP1) - AC4(L, ID1, IP) + BC4)
WC5(K) = BC5 + ID*(AC5(L, ID1, IP) - BC5) + DIP*(AC5(L, ID, IP1) - BC5 + ID)*(A waters 1670
1C5(L, ID1, IP1) - AC5(L, ID, IP1) - AC5(L, ID1, IP) + BC5)
WC6(K) = BC6 + ID*(AC6(L, ID1, IP) - BC6) + DIP*(AC6(L, ID, IP1) - BC6 + ID)*(A waters 1690
1C6(L, ID1, IP1) - AC6(L, ID, IP1) - AC6(L, ID1, IP) + BC6)

WAVS1230
WAVS1240
WAVS1250
WAVS1260
WAVS1270
WAVS1280
WAVS1290
WAVS1300
WAVS1310
WAVS1320
WAVS1330
WAVS1340
WAVS1350
WAVS1360
WAVS1370
WAVS1380
WAVS1390
WAVS1400
WAVS1410
WAVS1420
WAVS1430
WAVS1440
WAVS1450
WAVS1460
WAVS1470
WAVS1480
WAVS1490
WAVS1500
WAVS1510
WAVS1520
WAVS1530
WAVS1540
WAVS1550
WC7(K) = BC7+CID*(AC7(L, ID1, IP1)-BC7)+DIP*(AC7(L, ID1, IP1)-BC7)+DID*(AWAVS1710)
1C7(L, ID1, IP1)-AC7(L, ID1, IP1)-AC7(L, ID1, IP1)+BC7))
WC8(K) = BC8+DID*(AC8(L, ID1, IP1)-BC8)+DIP*(AC8(L, ID1, IP1)-BC8)+DID*(AWAVS1730)
1C8(L, ID1, IP1)-AC8(L, ID1, IP1)-AC8(L, ID1, IP1)+BC8))
WS0(K) = BS0+DID*(AS0(L, ID1, IP1)-BS0)+DIP*(AS0(L, ID1, IP1)-BS0)+DID*(AWAVS1750)
1S0(L, ID1, IP1)-AS0(L, ID1, IP1)-AS0(L, ID1, IP1)+BS0))
WS1(K) = BS1+DID*(AS1(L, ID1, IP1)-BS1)+DIP*(AS1(L, ID1, IP1)-BS1)+DID*(AWAVS1790)
1S1(L, ID1, IP1)-AS1(L, ID1, IP1)-AS1(L, ID1, IP1)+BS1))
WS2(K) = BS2+DID*(AS2(L, ID1, IP1)-BS2)+DIP*(AS2(L, ID1, IP1)-BS2)+DID*(AWAVS1810)
1S2(L, ID1, IP1)-AS2(L, ID1, IP1)-AS2(L, ID1, IP1)+BS2))
WS3(K) = BS3+DID*(AS3(L, ID1, IP1)-BS3)+DIP*(AS3(L, ID1, IP1)-BS3)+DID*(AWAVS1830)
1S3(L, ID1, IP1)-AS3(L, ID1, IP1)-AS3(L, ID1, IP1)+BS3))
WS4(K) = BS4+DID*(AS4(L, ID1, IP1)-BS4)+DIP*(AS4(L, ID1, IP1)-BS4)+DID*(AWAVS1850)
1S4(L, ID1, IP1)-AS4(L, ID1, IP1)-AS4(L, ID1, IP1)+BS4))
WS5(K) = BS5+DID*(AS5(L, ID1, IP1)-BS5)+DIP*(AS5(L, ID1, IP1)-BS5)+DID*(AWAVS1870)
1S5(L, ID1, IP1)-AS5(L, ID1, IP1)-AS5(L, ID1, IP1)+BS5))
WS6(K) = BS6+DID*(AS6(L, ID1, IP1)-BS6)+DIP*(AS6(L, ID1, IP1)-BS6)+DID*(AWAVS1890)
1S6(L, ID1, IP1)-AS6(L, ID1, IP1)-AS6(L, ID1, IP1)+BS6))
WS7(K) = BS7+DID*(AS7(L, ID1, IP1)-BS7)+DIP*(AS7(L, ID1, IP1)-BS7)+DID*(AWAVS1910)
1S7(L, ID1, IP1)-AS7(L, ID1, IP1)-AS7(L, ID1, IP1)+BS7))
WS8(K) = BS8+DID*(AS8(L, ID1, IP1)-BS8)+DIP*(AS8(L, ID1, IP1)-BS8)+DID*(AWAVS1930)
1S8(L, ID1, IP1)-AS8(L, ID1, IP1)-AS8(L, ID1, IP1)+BS8))
IF (K.EQ.2) GO TO 6
K = 2
L = IAA1
GO TO 5
6 BC0 = WCO(1)+DIA*(WC0(1)-WCO(1))
BC00 = WCO(1)+DIA*(WCO(2)-WCO(1))
BC1 = WC1(1)+DIA*(WC1(2)-WC1(1))
BC2 = WC2(1)+DIA*(WC2(2)-WC2(1))
BC3 = WC3(1)+DIA*(WC3(2)-WC3(1))
BC4 = WC4(1)+DIA*(WC4(2)-WC4(1))
BC5 = WC5(1)+DIA*(WC5(2)-WC5(1))
BC6 = WC6(1)+DIA*(WC6(2)-WC6(1))
BC7 = WC7(1)+DIA*(WC7(2)-WC7(1))
BC8 = WC8(1)+DIA*(WC8(2)-WC8(1))
BS0 = (WS0(1)+DIA*(WS0(2)-WS0(1)))*SALP
BS00 = (WS0(1)+DIA*(WS0(2)-WS0(1)))*SALP
BS1 = (WS1(1)+DIA*(WS1(2)-WS1(1)))*SALP
BS2 = (WS2(1)+DIA*(WS2(2)-WS2(1)))*SALP
BS3 = (WS3(1)+DIA*(WS3(2)-WS3(1)))*SALP
BS4 = (WS4(1)+DIA*(WS4(2)-WS4(1)))*SALP
BS5 = (WS5(1)+DIA*(WS5(2)-WS5(1)))*SALP
BS6 = (WS6(1)+DIA*(WS6(2)-WS6(1)))*SALP
BS7 = (WS7(1)+DIA*(WS7(2)-WS7(1)))*SALP
BS8 = (WS8(1)+DIA*(WS8(2)-WS8(1)))*SALP
SHIFT MOMENT CENTER FROM XREF TO C.G.

BC00 = BC00 - (XK-XREF)*PC0
BC3 = BC3 - (XK-XREF)*BC1
BC4 = BC4 - (XK-XREF)*BC2
BC6 = BC6 - (XK-XREF)*BC5
BS00 = BS00 - (XK-XREF)*BS0
BS3 = BS3 - (XK-XREF)*BS1
BS4 = BS4 - (XK-XREF)*BS2
BS6 = BS6 - (XK-XREF)*BS5

CALCULATE WAVE FORCES AND MOMENTS

FLC = BS1 - XK*G*(BS2 + BS0) - U*OM1*(-A33S*CK - AL*BS2)
FLS = BC1 - XK*G*(BC2 + BC0) + U*OM1*(-A33S*SK + AL*BC2)
FLM = BS3 - XK*G*(BS4 + BS0) - U*OM1*(-A33S*XXS*CK - BC2 - AL*BS4)
FLS = BC3 - XK*G*(BC4 + BC0) + U*OM1*(-A33S*XXS*SK - BS2 - AL*BC4)
FYC = XK*G*(BC5 + BC0) - U*OM1*(-A22S*SK + AL*BC5)
FYS = -XK*G*(BS5 + BS0) - U*OM1*(-A22S*CK - AL*BS5)
FRN = XK*G*(BC6 + BC0) + U*OM1*(-A22S*XXS*SK - BS5 - AL*BC6)
FRS = -XK*G*(BS6 + BS0) - U*OM1*(-A22S*XXS*CK - BC5 - AL*BS6)
FKC = XK*G*(BC7 + BC0) + U*OM1*(-A42S*SK + AL*BC8)
FKS = -XK*G*(BS7 + BS0) + U*OM1*(-A42S*CK - AL*BS8)

FZM(J) = FZM(J) - AA*(FZC*CT + FZS*ST)
FWM(J) = FWM(J) + AA*(FMC*CT + FMS*ST)
FYW(J) = FYW(J) - AA*(FYC*CT + FYS*ST)*SIGAM
FWM(J) = FWM(J) - AA*(FNC*CT + FNS*ST)*SIGAM
FKW(J) = FKW(J) - AA*(FKC*CT + FKS*ST)*SIGAM
FZX(J) = FZX(J) - 2.0*AA*RHD*G*BS*DSS*SK*CT

7 CONTINUE

C

IF (IMT, EQ. 2) GO TO 11
C

WAVE ELEVATION AROUND THE SIDEWALLS AND SEALS

DO 8 J=1,4
N = NSTA(J)
C
DO 8 K=1,N
ETA(J,K) = ETA(J,K) + SIN(XW*K*(-XX(J,K)*CGAM - YY(J,K)*SIGAM) + FT)*AA
C
8 CONTINUE
C
ETACG = ETACG + AA*SIN(FT)
N = NSTA(3)
C
DO 9 J=1, N
C
WAVS 21 90
WAVS 22 00
WAVS 22 10
WAVS 22 20
WAVS 22 30
WAVS 22 40
WAVS 22 50
WAVS 22 60
WAVS 22 70
WAVS 22 80
WAVS 22 90
WAVS 23 00
WAVS 23 10
WAVS 23 20
WAVS 23 30
WAVS 23 40
WAVS 23 50
WAVS 23 60
WAVS 23 70
WAVS 23 80
WAVS 23 90
WAVS 24 00
WAVS 24 10
WAVS 24 20
WAVS 24 30
WAVS 24 40
WAVS 24 50
WAVS 24 60
WAVS 24 70
WAVS 24 80
WAVS 24 90
WAVS 25 00
WAVS 25 10
WAVS 25 20
WAVS 25 30
WAVS 25 40
WAVS 25 50
WAVS 25 60
WAVS 25 70
WAVS 25 80
WAVS 25 90
WAVS 26 00
WAVS 26 10
WAVS 26 20
WAVS 26 30
WAVS 26 40
WAVS 26 50
WAVS 26 60
ARG = AA*COS(XWK*(-XX(3,J)*COGAM)+FT)
DETAX(J) = DETAX(J)-XWK*COGAM*ARG

9 CONTINUE

C N = NSTA(4)

DO 10 J=1,N
ARG = AA*COS(XWK*(-XX(4,J)*COGAM)+FT)
DETAX(J) = DETAX(J)-XWK*COGAM*ARG

10 CONTINUE

C WAVE PUMPING

X1 = XWK*XLBW*COGAM*0.5
X2 = XWK*XBFW*SIGAM*0.5
FTT = FT-XWK*XCPC*COGAM

11 CONTINUE

IF (TIME.EQ.0.0) GO TO 12
DVWDOT = (DVOLW-DVOLWO)/(TIME-TIMEO)

12 DVOLWO = DVOLW
TIMEO = TIME
IF (IMT.EQ.2) RETURN

C TOTAL WAVE FORCES AND MOMENTS

FXWAV = FXW(1)+FXW(2)
FYWAV = FYW(1)+FYW(2)
FZWAV = FZW(1)+FZW(2)+(FZW(2)-FZW(1))*YSW+FYWAV*ZBAR
FKWAV = KFW(1)+KFW(2)+(KFW(2)-KFW(1))*YSW+FYWAV*ZBAR
FMWAV = FMW(1)+FMW(2)-FXWAV*ZBAR
FNWAV = FNW(1)+FNW(2)+(FNW(2)-FNW(1))*YSW+FYWAV*ZBAR

IF (IWAVS.NE.ON) RETURN
WRITE (6,13) ((ETA(1,J), J=1,11), I=1,4), ETAGC,DVOLW,FXWAV,FYWAV,FZWAV
1AV,FKWAV,FMWAV,FNWAV

13 FORMAT (/10X,5HWAVES/63H WAVE ELEVATIONS AT CRAFT STATIONS RELATIVE TO CALM WATER (FT.)/14H PORT SIDEWALL/11F10.5/14H STBD SIDEWALL/11F10.5
1F10.5/1F10.5/1F10.5/25H WAVE ELEVATIONS/3100
31CN AT C.G. =F10.5,10X,43HPLNUM VOLUME LOST DUE TO WAVES (CU. FT.)/WAVS3110
4) =F15.5/10X,23H WAVES FX,FY,FZ,FK,FM,FN/6E15.4)

END
LIST OF REFERENCES


9. Private communication from F. Wilson to Professor A. Gerba, dated, 10 May 1974

10. Private communication from J. E. Blaloch to Professor A. Gerba, dated, 9 September 1975


INITIAL DISTRIBUTION LIST

1. Defense Documentation Center
   Cameron Station
   Alexandria, Virginia 22314

2. Library, Code 0212
   Naval Postgraduate School
   Monterey, California 93940

3. Associate Professor Alex Gerba, Jr., Code 52Gz
   Department of Electrical Engineering
   Naval Postgraduate School
   Monterey, California 93940

4. Professor George J. Thaler, Code 52Tr
   Department of Electrical Engineering
   Naval Postgraduate School
   Monterey, California 93940

5. Associate Professor Donald M. Layton, Code 57Ln
   Department of Aeronautical Engineering
   Naval Postgraduate School
   Monterey, California 93940

6. Mr. Arnold W. Anderson, Code PMS-304-31A.1
   Department of the Navy
   Surface Effect Ships Project Office, PM-17
   Bethesda, Maryland 20034
   (P. O. Box 34401)

7. Lieutenant John M. Boggio
   Naval Guided Missile School
   Dam Neck
   Virginia Beach, Virginia 23461

No. Copies

2
2
3
1
1
6
1
A study of pressure-volume rates and ple